

---

Entorno de experimentación para  
generación de indicaciones en interiores.

---



Proyecto Fin de Máster

Héctor Abraham Morillo Prieto

*Dirigido por:*

Pablo Gervás Gómez-Navarro, Gonzalo Méndez Pozo

Departamento de Ingeniería del Software e Inteligencia Artificial

Facultad de Informática

Universidad Complutense de Madrid

Junio 2013

Documento maquetado con T<sub>E</sub>X<sup>S</sup> v.1.0+.

# Entorno de experimentación para generación de indicaciones en interiores.

*Proyecto Fin de Máster*

**Ingeniería del Software e Inteligencia Artificial  
2013/6**

*Dirigida por*

**Pablo Gervás Gómez-Navarro, Gonzalo Méndez Pozo**

*Dirigido por:*

**Pablo Gervás Gómez-Navarro, Gonzalo Méndez Pozo**

**Departamento de Ingeniería del Software e Inteligencia  
Artificial**

**Facultad de Informática  
Universidad Complutense de Madrid**

**Junio 2013**

Copyright © Héctor Abraham Morillo Prieto

# Autorización

Héctor Abraham Morillo Prieto  
17 Febrero 2013

El/la abajo firmante, matriculado/a en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: GENERACIÓN DE INDICACIONES EN INTERIORES, realizado durante el curso académico 2012-2013 bajo la dirección de Pablo Gervás y Gonzalo Méndez en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

*A mis padres.*





# Agradecimientos

*A mis padres por todo su esfuerzo y  
dedicación.*

*A Pablo y Gonzalo, por toda su ayuda y  
colaboración.*

*A los alumnos que desinteresadamente  
participaron en el estudio piloto.*

*Gracias a todos.*



# Abstract

We designed and analyzed an experimental environment for generating indoor indications. It aims to be an aid to the study of expressions and language used by people when giving directions to others who do not know the path to follow to reach a destination. We have implemented an application that can load a 3D model with an integrated chat through which guide and guided can communicate, all this information is stored so that it can be reproduced and subsequently analyzed. This will generate a log of all information exchanged between the users and the route taken too. Log analysis can give very important information about the use of language to those interested in implementing a generated natural language instructions.

## Keywords

Instructions generation , natural language generation, 3D model, virtual environment, indications, experimentation, chat, guide, guided.



# Resumen

Se ha diseñado y analizado un entorno de experimentación para la generación de indicaciones en interiores. Pretende ser una ayuda para el estudio de las expresiones y el lenguaje utilizado por las personas a la hora de dar indicaciones a otras que no conocen la ruta a seguir para alcanzar un destino. Se ha implementado una aplicación capaz de cargar un modelo en 3D, con un chat integrado a través del cual guía y guiado pueden comunicarse, toda esa información se almacena de forma que pueda ser reproducida y analizada posteriormente. Para ello se genera un log con toda la información intercambiada entre los usuarios así como el recorrido realizado. El análisis de ese log puede dar datos muy importantes sobre la utilización del lenguaje a aquellas personas interesadas en implementar un generador de indicaciones en lenguaje natural.

## Palabras clave

Generación de instrucciones, generación de lenguaje natural, modelo 3D, entorno virtual, indicaciones, experimentación, chat, guía, guiado.



# Índice

<b>Autorización</b>	<b>V</b>
<b>Agradecimientos</b>	<b>IX</b>
<b>Abstract</b>	<b>XI</b>
<b>Resumen</b>	<b>XIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	2
1.2. Objetivos . . . . .	4
1.2.1. Objetivos tecnológicos . . . . .	4
1.2.2. Objetivos científicos . . . . .	5
1.3. Fases de desarrollo . . . . .	5
<b>2. Trabajos Relacionados</b>	<b>7</b>
2.1. Entorno virtual 3D multiusuario para simulación de escenarios de evacuación . . . . .	7
2.2. Modelado de evacuación de multitudes mediante agentes y transcripción de comportamientos . . . . .	8
2.3. Simulaciones de emergencia en entornos 3D . . . . .	9
2.4. A Model for Human Readable Instruction Generation Using Level-Based Discourse Planning and Dynamic Inference of Attributes Disambiguation . . . . .	10
2.5. Generación de Lenguaje Natural . . . . .	11
2.5.1. Historia de la Generación de Lenguaje Natural . . . . .	12
2.5.2. Fases de la Generación de Lenguaje Natural . . . . .	13
2.6. Algunos sistemas generadores de lenguaje natural . . . . .	14
2.6.1. WeatherReporter . . . . .	15
2.6.2. FOG . . . . .	15
2.6.3. STOP . . . . .	15
2.6.4. ERMA . . . . .	17

2.6.5. MEXICA . . . . .	18
2.7. Entorno virtual o modelo 3D . . . . .	19
<b>3. Planificación, Diseño e Implementación de la Aplicación</b>	<b>23</b>
3.1. Primer prototipo . . . . .	23
3.2. Segundo prototipo . . . . .	24
3.3. Tercer prototipo . . . . .	27
3.4. Técnicas y herramientas . . . . .	28
3.4.1. OGRE 3D . . . . .	28
3.4.2. CEGUI . . . . .	29
3.4.3. Sockets - Chat . . . . .	30
<b>4. Diseño de los experimentos</b>	<b>33</b>
4.1. Introducción . . . . .	33
4.2. Descripción . . . . .	35
4.3. Primer estudio piloto . . . . .	37
<b>5. Resultados</b>	<b>39</b>
5.1. Resultado del experimento piloto . . . . .	42
<b>6. Conclusiones y trabajo futuro</b>	<b>45</b>
6.1. Conclusiones . . . . .	45
6.2. Trabajo futuro . . . . .	46
6.2.1. Entorno de experimentación . . . . .	46
6.2.2. Generación de logs y comunicación por red . . . . .	47
6.2.3. Reproducción de sesiones anteriores . . . . .	48
6.2.4. Diseño y realización de experimentos . . . . .	48
<b>I Apéndices</b>	<b>51</b>
<b>A. Manual de usuario (Inicio rápido)</b>	<b>53</b>
A.1. Inicio del servidor chat . . . . .	53
A.2. Inicio de la aplicación . . . . .	54
A.3. Navegación por la aplicación . . . . .	56
<b>B. Posible rutas para experimento</b>	<b>59</b>
B.1. Ruta 1 . . . . .	59
B.2. Ruta 2 . . . . .	60
B.3. Ruta 3 . . . . .	60
B.4. Ruta 4 . . . . .	61
B.5. Ruta 5 . . . . .	61

---

<b>C. Ejemplos de Logs</b>	<b>63</b>
C.1. Log generado por un usuario en modo Guiado . . . . .	63
C.2. Log generado por un usuario en modo Guía . . . . .	65
<b>Bibliografía</b>	<b>69</b>



# Índice de figuras

2.1. Captura de la aplicación del proyecto: Entorno virtual 3D multiusuario para simulación de escenarios de evacuación . . . . .	8
2.2. Captura de la aplicación y la representación en 2D . . . . .	9
2.3. Comparación entre «paso a paso» e instrucciones humanas . . . . .	11
2.4. Fases de la GLN . . . . .	14
2.5. Resumen del tiempo para Febrero de 1995 . . . . .	15
2.6. Entrada de FOG: Sistema meteorológico sobre Canada . . . . .	16
2.7. Salida de FOG . . . . .	16
2.8. Entrada del sistema STOP . . . . .	17
2.9. Salida del sistema STOP . . . . .	18
2.10. Fragmento generado por MEXICA . . . . .	20
2.11. Imagen de la réplica en Second Life del centro de estudios localizado campus universitario de Saint Leo, Florida. . . . .	21
3.1. Primer prototipo de la aplicación . . . . .	24
3.2. Servidor chat desarrollado en la segunda iteración . . . . .	25
3.3. Implementación del chat en el segundo prototipo . . . . .	26
3.4. Log generado en una ejecución de la aplicación . . . . .	26
3.5. Mapa 2D utilizado en el proyecto de Elena Núñez González . . . . .	28
3.6. Logotipo Ogre 3D . . . . .	29
3.7. Logotipo librería CEGUI . . . . .	30
5.1. Ventana de selección para el inicio de la aplicación . . . . .	40
5.2. Aplicación ejecutando en modo guía . . . . .	40
5.3. Aplicación ejecutando en modo guiado . . . . .	41
5.4. Servidor de chat . . . . .	41
5.5. Captura generada por la aplicación . . . . .	42
6.1. Ejemplo de minimapa dentro de un videojuego . . . . .	47
A.1. Servidor chat una vez que se ha iniciado correctamente . . . . .	53
A.2. Configuración inicial aplicación . . . . .	55

A.3. Vista general de la aplicación navegando por el modelo 3D . . .	55
A.4. Vista general de la aplicación con el chat . . . . .	56
C.1. Captura de imagen generada en una de las instrucciones - 28-5-13-14-1-51.jpg . . . . .	67

# Capítulo 1

## Introducción

En la actualidad los sistemas de guiado son prácticamente indispensables en el día a día de muchas personas, ya sea por motivos laborales o simplemente por ocio. Estos sistemas de guiado suelen aportar una valiosa información al usuario gracias a imágenes, flechas u otros tipos de indicaciones visuales. También son capaces de generar indicaciones audibles, aunque estas, en muchas ocasiones, solo son representaciones simples de las indicaciones gráficas.

El principal problema de las indicaciones anteriormente mencionadas es el lenguaje utilizado por la máquina para dar las instrucciones. Generalmente son indicaciones diferentes a las que daría cualquier persona ya que utilizan otras formas de expresión y otras referencias diferentes a las que utilizaríamos nosotros, por ejemplo: «Continúe recto durante cien metros, después gire a la izquierda», el problema en indicaciones de este tipo reside en que una persona no es capaz de medir por sus propios medios y de forma precisa distancias exactas y además no estamos acostumbrados a recibir ese tipo de indicaciones.

Los inconvenientes mencionados se agravan en el momento en que nos encontramos en el interior de edificios, donde sistemas como el GPS pierden su funcionalidad. Además, ese tipo de entornos son mucho más complejos y las instrucciones audibles basadas en distancias mucho más difíciles de interpretar. Lo ideal sería que las mismas máquinas que son capaces de generar la ruta que debemos seguir para llegar al punto deseado desde un punto de origen, fueran capaces de generar instrucciones y expresarlas de forma natural para los usuarios. Un ejemplo de estas sería «Continúa por este pasillo hasta el final y entra por la puerta azul que verás a tu derecha». Instrucciones similares a la anterior son mucho más sencillas de interpretar por las personas ya que se basan en percepciones que van adquiriendo a medida que realizan el recorrido.

Generar instrucciones en lenguaje natural es muy complicado ya que los sistemas deberían ser capaces de identificar objetos o partes relevantes del entorno a través de las cuales definir las instrucciones para un usuario. Ade-

más, estas instrucciones se deben crear siguiendo una estructura que facilite su comprensión. Para ello, sería necesario tener una idea clara de la forma en que una persona da indicaciones a otra y en qué nos basamos. En este punto es donde radica la dificultad principal de la generación de instrucciones en lenguaje natural: es difícil identificar buenas soluciones para determinar las formas de expresión de utilidad. Por ello es necesario la experimentación y un entorno propicio para desarrollar dichos experimentos.

La generación de este tipo de instrucciones puede ser, además, muy útil a la hora de guiar a persona con discapacidad, ya que son más fácilmente comprensibles y adaptables a las necesidades de cada persona, es decir, un sistema podría ser capaz de identificar de qué forma se pueden dar mejores indicaciones para determinadas personas. Por ejemplo, para una persona daltónica las instrucciones no deberían basarse en colores, sería necesario utilizar otros recursos.

Para todo lo mencionado anteriormente y para solucionar algunas de las necesidades que plantea la generación de instrucciones en lenguaje natural, se plantea este proyecto con objetivo de facilitar el análisis del lenguaje natural y la realización de experimentos para la adquisición de datos relacionados con el lenguaje y las formas de expresión.

## 1.1. Motivación

Poder implementar un sistema que sea capaz de realizar una generación de instrucciones de forma autónoma requiere, como primer paso a realizar, la recopilación de gran cantidad de información sobre cómo las personas dan indicaciones a otras personas en función del contexto en el que se encuentren. Después, una vez que disponemos de esa información, llegaría la implementación propiamente dicha del generador de instrucciones.

La base del estudio y del proyecto radica en el desarrollo de una aplicación de escritorio que permita obtener la mayor información posible sobre como las personas dan indicaciones a otras para alcanzar un determinado objetivo. La información que nos interesa obtener a través de la aplicación es los recursos del lenguaje que son utilizados para la generación de instrucciones, es decir, si la persona que está dando las indicaciones se basa en los objetos que se encuentran en el recorrido, en los colores, en la formas, o si se utilizan las distancias con los objetos u otras formas de expresión para tratar de explicar la forma de alcanzar el lugar de destino. También es interesante poder estudiar cómo esas indicaciones se realizan de diferente modo en función de la situación en la que se encuentren ambos interlocutores. Seguramente la forma de expresar las ideas será muy diferente cuando nos encontramos en interiores o en lugares cerrados, si la persona que necesita la información está conduciendo o andando, o por ejemplo si las indicaciones se están realizando de forma telefónica, donde encontraremos expresiones como «¿Estás viendo

la puerta azul?», a través de las cuales el guía trata de asegurarse de que la otra persona se encuentra donde él piensa.

Este campo de análisis es muy extenso, por lo que, en una primera aproximación, nos centraremos en las indicaciones realizadas en el interior de edificios, de forma que una vez realizado el análisis podamos establecer unas pautas para la generación de indicaciones en lenguaje natural.

Una opción para realizar el análisis de esas formas de expresión comentadas anteriormente podría haber sido realizar una serie de experimentos en un lugar real, donde las personas participante en la prueba deberían realizar los movimientos a través de las indicaciones de sus guías. Esto plantea ciertos problemas:

- Almacenamiento de las indicaciones para su posterior análisis: Sería necesario realizar una grabación con las indicaciones del guía o pedirle que las realizará por escrito ya que esta información es fundamental para su posterior análisis.
- Temporal: Para poder realizarlo sería necesario una gran cantidad de tiempo, tanto de los investigadores como de las personas que se sometían a las pruebas. Al ser necesario mayor cantidad de tiempo para cada una de las pruebas, sería posible obtener un menor número de muestras y por tanto la información tendría menos validez.
- Almacenamiento del recorrido realizado y de la utilidad de las indicaciones: Es necesario comprobar no sólo la forma de guía para dar las indicaciones, sino la precisión y validez de las mismas para la persona que está siendo guiada.

Para solucionar todos estos problemas lo que se plantea es utilizar un modelo virtual en 3D de un edificio, concretamente el de la Facultad de Informática de la UCM, de forma que todos esos problema sean resueltos a través del desarrollo de una aplicación para tal efecto.

Por tanto, con el objetivo de obtener un amplio conjunto de datos se decide implementar un modelo de la facultad de informática de la Universidad Complutense de Madrid (UCM) en 3D, dentro del cual un usuario pueda navegar por su interior. De esta forma es posible la realización simultánea de los experimentos con un gran número de personas, con la única limitación de los equipos hardware disponibles. Partiendo de este modelo, podemos diseñar una serie de experimentos en los que las personas involucradas deben expresar una serie de indicaciones a otros para conseguir un objetivo, podemos almacenar toda esa información generada y analizarla.

## 1.2. Objetivos

En este apartado, se van a desarrollar los principales objetivos que se pretende cumplir con este proyecto. Para tratarlos de una forma más clara se van a dividir en dos apartados: el primero de ellos hace referencia a los objetivos tecnológicos que se deben de cumplir, es decir, que características ha de tener el desarrollo software realizado. Por otro lado, el segundo apartado hace referencia a los objetivos que se pretenden cumplir a través de los experimentos desarrollados

### 1.2.1. Objetivos tecnológicos

En este apartado se van a especificar los objetivos tecnológicos que debe cumplir la aplicación desarrollada. Como ya ha sido comentado, la aplicación trata de ser una ayuda a la adquisición de datos sobre el lenguaje natural. Los objetivos son los siguientes:

- Desarrollo de un entorno de experimentación que permita la carga de un modelo 3D y la realización de experimentos dentro del mismo. El sistema implementado tiene que estar orientado al desarrollo de pruebas relacionadas con el análisis de lenguaje natural, concretamente con la generación de indicaciones. Es necesario que el entorno desarrollado sea fácil de usar por cualquier persona, tenga los conocimientos informáticos que tenga. Es posible que el entorno de experimentación tenga que realizar comunicación de red, por lo que dentro de este objetivo es necesario verificar el correcto funcionamiento de la comunicación así como la robustez general del sistema.
- Implementación de un sistema de generación de logs que contengan información suficiente para regenerar todos los experimentos previamente realizados. El sistema de logs tiene una importancia fundamental ya que todo el análisis posterior de los experimentos se basa en él, es necesario comprobar que toda la información es correctamente almacenada. Otro punto importante es comprobar que en caso de que el entorno de experimentación sufra un error, no se perderá la información ya almacenada.
- Relacionado con el objetivo anterior se plantea como una necesidad implementar un sistema que ayude de forma visual al posterior análisis de los resultados, de forma que en los puntos claves del experimento se pueda almacenar el contexto visualizado a través del entorno. Se podría plantear como un objetivo complementario al anterior, ya que pretende ser una ayuda a la información generada por él.

### 1.2.2. Objetivos científicos

Vamos a enumerar los objetivos que se deben de cumplir con respecto a los experimentos realizados en base a la aplicación desarrollada, es decir, los puntos que se deben cumplir para verificar que la aplicación cumple la funcionalidad para la que ha sido desarrollada.

- Obtener información sobre las expresiones utilizadas a la hora de dar indicaciones así como el contexto en el que se utiliza cada una de ellas. Detectar similitudes entre diferentes expresiones que busquen un mismo objetivo pero indicado de formas diferentes.
- Detectar que expresiones son más efectivas, es decir, que expresiones tiene mayor tasa de acierto a la hora de ser realizadas por el usuario guiado y cuales menos. Esto permite conocer qué tipo de expresiones deberían ser utilizadas por un sistema generador de instrucciones a través de lenguaje natural.

En resumen, la herramienta desarrollada debe facilitar la realización de experimentos científicos relacionados con la generación de instrucciones en lenguaje general, siendo una forma de analizar y evaluar las formas de expresión utilizadas por las personas.

## 1.3. Fases de desarrollo

Para el desarrollo del proyecto planteado, se proponen una serie de fases que se indican a continuación. Como se puede observar, las fases iniciales hacen referencia al desarrollo tecnológico mientras que las últimas hacen referencia al desarrollo científico. Las fases son las siguientes:

1. Desarrollo de un modelo 3D: Como se ha comentado anteriormente, el modelo en 3D de la Facultad de Informática de la UCM es la base del proyecto y de los experimentos que van a ser realizados. Partiendo de una representación de la Facultad de Informática previa al desarrollo de este proyecto, se propone desarrollar una aplicación que cargue dicho modelo y permita a un usuario navegar de forma libre por el mismo.
2. Implementación de un chat de comunicación en el modelo anterior: Se pretende añadir a la aplicación anterior, encargada de la carga del modelo en 3D, un chat de comunicación, la idea es que un usuario sea el que debe realizar un recorrido desde un punto a otro, y pueda solicitar ayuda y recibir indicaciones a través del chat para conseguir finalizar el recorrido.
3. Log de todos con la información necesaria de la navegación por el modelo: Una parte fundamental del desarrollo del proyecto es el almacenamiento de datos para su posterior análisis. Se requiere almacenar

todas las indicaciones que son necesarias por un usuario para realizar un recorrido. Además, para completar esa información, se realizarán capturas de pantalla e incluso vídeos completos de cada uno de los movimientos que se realizan dentro del modelo. También es necesario almacenar cada una de las posiciones por las que va moviéndose un usuario dentro del modelo de forma que se pueda recrear cada uno de los pasos que ha realizado.

4. Diseño de los experimentos: Se plantea realizar los experimentos por parejas de usuarios, donde uno de ellos deberá realizar un recorrido dentro del modelo con las indicaciones del otro, utilizando para ello el chat de comunicación que se ha implementado dentro de la aplicación. Una vez que un usuario ha sido capaz de realizar varios recorridos se cambian las posiciones y ahora la persona que anteriormente estaba indicado debe ser la que realice los recorridos. Es muy importante realizar tantos recorridos y experimentos como sea posible, ya que eso enriquecerá en gran medida los resultados obtenidos y la precisión de los resultados.
5. Realización de los experimentos: Se realizarán los experimentos diseñados en el punto anterior con la mayor cantidad de gente posible, se procurará de dar la información suficiente a los participantes pero teniendo la precaución de no influenciar el comportamiento posterior, es decir, todos los participantes deben conocer cuál es su objetivo, llegar a un determinado objetivo o bien guiar a otra persona hasta el objetivo, pero no se debe condicionar a estas personas de forma que las indicaciones o preguntas no se expresen de forma espontánea, como lo harían en una situación real. Esto es de vital importancia, ya que de ello depende la calidad de los resultados obtenidos y por consiguiente de todo aquello que sea generado a partir de ellos.
6. Análisis de los resultados obtenidos: Una vez realizados los experimentos es necesario analizar todos los logs generados para obtener patrones a la hora de la generación de instrucciones de forma que posteriormente se puedan generar instrucciones siguiendo esos mismos patrones.

Después de realizar los pasos anteriormente citados, tendremos información suficiente para comprender cómo los seres humanos realizamos la generación de instrucciones y, por tanto, de qué forma somos capaces de interpretarlas mejor. A partir de este momento sería posible intentar implementar un generador de instrucciones autónomo y automático, de forma que el sistema, analizando el recorrido que se debe de explicar y las características o elementos que pueden ser destacables en las instrucciones genere las instrucciones. Esto queda fuera del marco de este proyecto, donde se estudia, implementa y prueba un entorno de experimentación.

## Capítulo 2

# Trabajos Relacionados

### 2.1. Entorno virtual 3D multiusuario para simulación de escenarios de evacuación

Proyecto final de carrera de la titulación de Ingeniería Informática [Cruz Ruiz et al. (2011)], realizado en la Universidad Complutense de Madrid (UCM) que desarrolló un entorno virtual de la Facultad de Informática. El objetivo del proyecto era permitir realizar simulaciones de incendios y de la evacuación del edificio. Para ello, los autores del proyecto desarrollaron un aplicación utilizando el lenguaje PyOgre que permite realizar simulaciones de emergencias en dicho entorno.

Una de las características más llamativas de la aplicación es la conexión en red. Se trata de un entorno multiusuario, cliente-servidor. Esta característica permite la conexión de varios usuarios que realizar acciones diferentes dentro del sistema, existen dos roles dentro de la aplicación, por un lado personas que se encuentran dentro de la facultad y deben conseguir salir de ella en el menor tiempo posible y sin poner en riesgo su integridad y por otro lado los denominados "guías". Estos son los encargados de responder a las peticiones de ayuda de los usuarios y darles las indicaciones necesarias. En este punto es donde entra en juego la capacidad de conexión a través de la red, ya que permite a través del chat implementado dentro de la aplicación comunicarse entre los usuarios. Ver Figura 2.1.

Esta característica es muy similar a la planteada por nuestro proyecto para realizar el estudio sobre la comunicación de indicaciones entre personas. Por ello, inicialmente se pensó en reutilizar esta funcionalidad en la aplicación de apoyo al proyecto. El principal problema que se planteó fue la tecnología utilizada (PyOgre), ya que no tiene un gran soporte por parte de la comunidad y actualmente su evolución se encuentra estancada.

Aunque finalmente, por motivos técnicos, no se ha podido reutilizar la parte del desarrollo realizado en este proyecto que podía ser interesante para



Figura 2.1: Captura de la aplicación del proyecto: Entorno virtual 3D multiusuario para simulación de escenarios de evacuación

nuestro proyecto, ha sido importante conocer como se había planteado la solución de algunos de los problemas de comunicación en red y como habían tratado de solucionarlos los autores del proyecto.

## 2.2. Modelado de evacuación de multitudes mediante agentes y transcripción de comportamientos

Este proyecto fue presentado como Trabajo Fin de Máster en la Universidad Complutense de Madrid [Nuñez González (2011)]. En él se ha diseñado y analizado un sistema multi-agente que simula el comportamiento de las personas involucradas en una evacuación de incendios en un edificio. Se han simulado tres tipos de comportamiento: El líder, que ayuda a los demás a salir del escenario, el dependiente, sin capacidad para salir por cuenta propia, y que necesita que un líder le ayude, y el independiente, con capacidad para evacuar por sí mismo, pero que en ocasiones decide poner su vida en peligro saltándose las normas de evacuación. La simulación produce un extenso log con los eventos sucedidos. Se ha creado un sistema, FromLog2StoryTelling, que trata el log, lo resume y modifica para generar un informe con los sucesos más importantes ordenados, y preparado para traducirse a lenguaje natural.

En este trabajo se realiza una representación en dos dimensiones de una de las plantas sobre la que se realiza una simulación de un incendio dentro del edificio, ver Figura 2.2. La aplicación permite ver como evoluciona y como

las personas deben ir abandonando el edificio según los roles mencionados anteriormente.



Figura 2.2: Captura de la aplicación y la representación en 2D

Se fija como objetivo de la aplicación del proyecto en el que nos encontramos realizar una función que permita capturar los movimientos que hace el usuario por el entorno en 3D y ser capaces de obtener una relación con la coordenadas 2D del proyecto [Nuñez González (2011)]. De esta forma, se podrá visualizar en los mapas utilizados el recorrido realizado por los usuarios a los que realicemos los experimentos. Esto puede tener gran utilidad a la hora de estudiar los resultados obtenidos a partir de los experimentos, así como a la hora de diseñar el sistema de GLN que será el encargado de producir las indicaciones en base a elementos destacados del modelo.

### 2.3. Simulaciones de emergencia en entornos 3D

Proyecto final de carrera de la titulación de Ingeniería Informáticos [Martínez del Camino (2012)], realizado en la Universidad Complutense de Madrid (UCM), que desarrolló un entorno virtual de la Facultad de Informática tomando como base el proyecto anteriormente mencionado "Modelado de evacuación de multitudes mediante agentes y transcripción de comportamientos", la aplicación desarrollado aporta un nuevo punto de vista a la simulación de incendios, ya que permite moverse libremente por el edificio durante los momentos cruciales de la evacuación sin perder detalle de todo lo que acontece. Esta herramienta tiene como cometido extraer datos que puedan servir para mejorar tanto las instalaciones, como los protocolos de evacuación de forma que se minimicen los daños y, sobre todo, las víctimas en caso de incendio.

Para la realización de este trabajo, se desarrolló un aplicación que, entre otras funciones, cargaba un modelo de la facultad de informática en 3D,

sobre el que se realizaba toda la simulación de la emergencia. Inicialmente se propuso reutilizar parte del código implementado, concretamente la parte encargada de cargar el modelo en 3D. Al tratar de construir la nueva aplicación partiendo de esta base de código, aparecieron dos problemas relacionados entre sí. Por un lado, el modelo 3D que se utilizaba en la aplicación estaba anticuado y poseía pocos detalles, por lo que era necesario reemplazarlo por otro modelo más moderno que teníamos a nuestra disposición. Al tratar de realizar esta actualización del modelo, surgió el segundo problema, la versión de Ogre utilizada estaba anticuada y no permitía cargar el nuevo modelo. Finalmente fue necesario implementar la aplicación con la última versión de Ogre disponible (Febrero de 2013), fue de gran utilidad conocer la implementación realizada por el autor de este proyecto aunque, finalmente, no se pudiera reutilizar todo el código previsto.

## 2.4. A Model for Human Readable Instruction Generation Using Level-Based Discourse Planning and Dynamic Inference of Attributes Disambiguation

Se utiliza este trabajo [Dionne et al. (2009)] junto con [León et al. (2009)], realizado en la Universidad Complutense de Madrid, como artículos de referencia, ya que analizan de forma clara cómo son las instrucciones generadas por los sistemas de guiado actuales y qué se espera conseguir para mejorar los mismos. Lo que trata de explicar la parte inicial del artículo es cómo la gran mayoría de sistemas de guiado actuales realiza una planificación «paso a paso», es decir, el sistema genera una instrucción para cada uno de las indicaciones cuando lo realmente interesante sería dar una serie de instrucciones más humanizadas, que más claro con la Figura «2.3».

Como se puede suponer las instrucciones paso a paso serían algo similar a los siguiente: «Gira a la izquierda, gira a la derecha, avanza, gira a la derecha...», mientras que una serie de instrucciones dadas por un humano serían mucho más sencillas y fáciles de interpretar «Sal de la habitación».

Además de esto, este trabajo presenta un modelo para la generación de este tipo de instrucciones, basado en el conocimiento del entorno y la meta que se desea alcanzar. Por ello y para poder generar instrucciones de forma adecuado es necesario conocer como se expresan los seres humanos en las diferentes situaciones de guiado en las que se pueden encontrar.

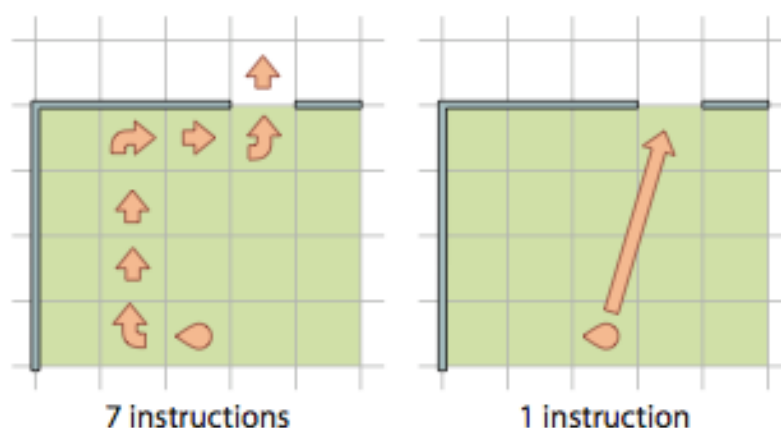


Figura 2.3: Comparación entre «paso a paso» e instrucciones humanas

## 2.5. Generación de Lenguaje Natural

El procesamiento de lenguaje natural (PLN) se suele dividir en dos partes claramente diferenciadas, por un lado encontramos la comprensión y el entendimiento, y por el otro, la generación propiamente dicha. Por tanto, en el PLN se estudia el proceso de comprensión del lenguaje así como de crear agentes capaces de interpretar correctamente un texto, por otro lado también se estudia la generación de textos en lenguaje natural mediante agentes artificiales.

La generación de lenguaje natural (GLN) o en inglés *Natural Language Generation (NLG)*, es el área encargada del diseño, desarrollo e implementación de sistemas capacitadas para generar de forma automática textos expresados en lenguaje natural. Como normal general, los sistemas de GLN se encargan de generar resúmenes, descripciones y otros tipos de textos, bien sean escritos o de voz a partir de información que se encuentra almacenada de forma no lingüística, tratando de expresar dicha información de forma más asequible para entenderse por las personas. Un general tiene la complicada tarea de decidir que palabras y estructuras sintácticas debe utilizar en cada comento, en función de la información que desee expresar y el idioma utilizado.

La generación de textos en lenguaje natural es un proceso que toma ciertos datos por entrada, en nuestro caso el recorrido que debe realizar una persona y que retorna texto como salida. A medida que la generación de lenguaje natural fue evolucionando se desarrollaron diferentes teorías, en ocasiones incompatibles, sobre la forma en la que se debería descomponer el proceso de GLN, por ello Reiter y Dale en [Reiter y Dale (2000)] definen un diseño que trata de estandarizar el proceso y proporciona una especificación abstracta de los elementos de entrada y salida de los sistemas de GLN.

Su arquitectura se utilizó como diseño base y ha sido adoptada como una arquitectura clásica para esta clase de sistemas (ver 2.5.2).

### 2.5.1. Historia de la Generación de Lenguaje Natural

Inicialmente se concebía la GLN como un campo sencillo, esto era debido a que la mayoría de las aplicaciones necesitaba únicamente, generar textos muy simples. Los textos generados, expresaban de forma básica aquello que se pretendía comunicar y aunque era claramente visible su origen computacional era común (incluso hoy sigue siéndolo) conformarse con esos resultados. La complejidad real del problema de GLN se descubrió cuando fue necesaria la generación de textos con la calidad de un escritor humano, fue entonces cuando la comunidad científica vio que los temas que había que tratar eran fundamentalmente los siguientes: representar lo que se debe decir y para qué se debe decir, seleccionar los conocimientos que deben ser expuestos, estructurar el texto de la forma más adecuada en cada momento para alcanzar los objetivos comunicativos, contemplar fenómenos lingüísticos difíciles de modelar, seleccionar la palabra o palabras apropiadas en cada contexto, dotar a los textos de variedad, mantener la cohesión y la coherencia, etc.

La GLN como la conocemos actualmente comenzó a mediados del siglo XX centrada en los requerimientos de los sistemas de traducción entre diferentes idiomas, es un campo de investigación bastante joven y en el que queda mucho terreno por recorrer debido a la dificultad para generar texto de calidad cercana a la de un humano.

En la década de 1970 se realizaron algunas investigaciones relacionadas con la GLN que permitieron sentar los principales temas de desarrollo e influencia de la GLN, también permitieron verificar que la GLN no debe entenderse como el proceso inverso de la CLN ya que cada uno de los procesos presenta dificultad y características diferentes y no complementarias. Algunas de las investigaciones pueden ser la de Goldman [1975] o las de Davey [1978].

Ya en los años ochenta, la investigación sobre la GLN experimentó un notable avance, alejándose de los sistemas monolíticos que pretendían ejecutar el proceso completo de generación, para dividir el problema en partes más pequeñas y claramente diferenciadas, como la realización sintáctica, la elección léxica, y la estructuración del texto. Uno de los ejemplos de esta época es el sistema ÓSCAR [Cohén y Perrault (1979)], que planificaba los actos de habla (informar o pedir) y el contenido proposicional de las respuestas a las consultas de los usuarios. Por otro lado, investigadores como McKeown [McKeown (1985)] o Mann y Thompson [Mann y Thompson (1988)] tuvieron gran trascendencia con sus trabajos doctorales sobre la evolución de este campo. También comenzaron a surgir los primeros congresos y conferencias entre las que destaca la «International Workshop on Natural Language Generation», celebrado por primera vez en 1983.

En los años noventa e incluso en la actualidad, se continúan muchos de los estudios e investigaciones iniciados en la década de los ochenta, además ha crecido la motivación por la combinación de la generación de textos y otras posibilidades de representación de la realidad como pueden ser: gráficos o sonidos. También durante la década de los noventa surgieron nuevos sistemas de GLN como por ejemplo FOG [Goldberg et al. (1994)], que fue uno de los sistemas más famosos y el primero en utilizarse de forma comercial. FOG es un asistente para generar reportes del clima que permite al usuario observar un mapa en dos dimensiones y realizar un reporte en inglés o francés para un área concreta seleccionada por el usuario.

En la actualidad la GLN tiene gran cantidad de problemas enunciados, pero muchos aun sin resolver. Este proyecto tratará de profundizar en la generación de indicaciones basándose en elementos conocidos e identificativos.

### 2.5.2. Fases de la Generación de Lenguaje Natural

La GLN, normalmente, se encuentra dividida en varias fases que se ejecutan de forma sucesiva y que refinarán el contenido original (información de entrada) hasta generar el texto en el idioma natural escogido. Una de las estructuras mas extendidas es la de Reiter y Dale (1997), que fue ampliada y modificada por ellos mismos en el año 2000 [Reiter y Dale (2000)], las etapas de las que tratan en sus trabajos son las que se indican a continuación:

- *Planificación del texto*, consiste en seleccionar de toda la información de entrada disponible, aquella que se desea expresar a través del lenguaje natural, en función del objetivo de salida. Se puede dividir en dos suboperaciones.
  - *Determinación del contenido*, decidir de qué elementos han de ser expresados, y cuáles no deben tenerse en cuenta. Esta etapa es muy dependiente del dominio, y no es posible una generalización.
  - *Planificación del discurso*, consiste en realizar una planificación a alto nivel sobre en qué orden se van a relatar las diferentes partes de la información de entrada extraídas en la etapa anterior y que relaciones guardan entre sí.
- *Planificación de las frases (Microplanificación)*, recibe como entrada la información procesada, filtrada y ordenada por la etapa anterior de "Planificación del discurso", procesa esos mensajes para generar frases más próximas a la creadas por los humanos, los procesos que realiza son:
  - *Agregación*, consiste en unificar frases inicialmente separadas, que por su contenido pueden estar escritas en una única elocución

- *Lexicalización*, etapa en la cual se decide qué palabras y frases específicas del lenguaje objeto van a ser usadas para describir los hechos.
  - *Generación de expresiones de referencia*, describir cada concepto fundamental del texto generado en función de su colocación global dentro del conjunto creado.
- *Realización Lingüística*, finalmente, enlaza el discurso haciendo la flexión gramatical de las partes, y establece una ortografía correcta. Engloba dos tipos de operaciones:
- *Realización sintáctica y morfológica*, atendiendo a la flexión de las reglas gramaticales.
  - *Realización ortográfica*, realiza el ajuste del texto finalmente generado, a las reglas ortográficas de la lengua natural en la que se genera el mensaje.

En la Figura 2.4 se trata de expresar de forma gráfica estas fases de la generación.

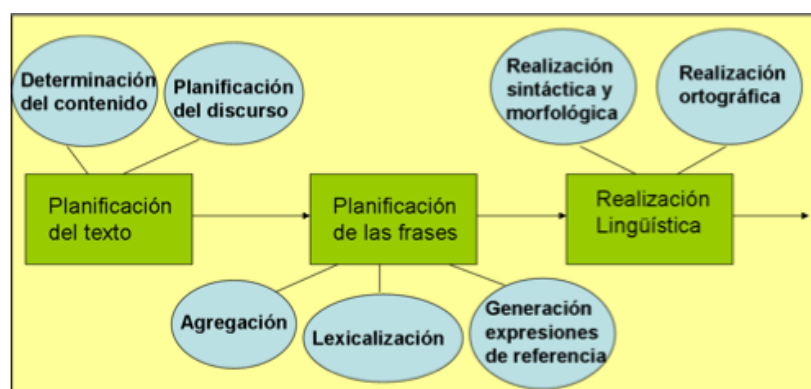


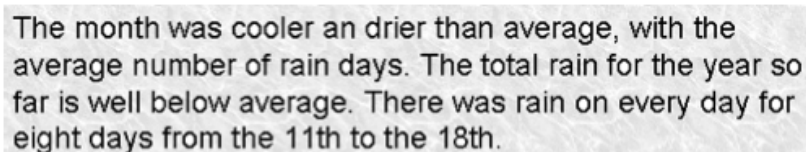
Figura 2.4: Fases de la GLN

## 2.6. Algunos sistemas generadores de lenguaje natural

En este apartado vamos a comentar, de forma breve, algunos sistemas GLN destacados así como su funcionalidad y sus características particulares, de forma que el lector pueda conocer los usos generales que han tenido hasta hoy este tipo de sistemas.

### 2.6.1. WeatherReporter

Aplicación implementada en Perl y XML que genera informes retrospectivos en Inglés del tiempo sobre periodos de duración de un mes. Presenta como entrada, un conjunto de datos recogidos de forma automática por diferentes sensores meteorológicos (temperatura y lluvia). La salida son uno o dos párrafos de longitud en función de la naturaleza de los datos recogidos, ver Figura 2.5.



The month was cooler and drier than average, with the average number of rain days. The total rain for the year so far is well below average. There was rain on every day for eight days from the 11th to the 18th.

Figura 2.5: Resumen del tiempo para Febrero de 1995

### 2.6.2. FOG

El sistema FoG (Forecast Generator) [Goldberg et al. (1994)] permite generar un informe meteorológico, a partir de una predicción de ciertos valores que afectan en el clima como pueden ser la velocidad del viento, las precipitaciones y otros factores que varían según la región. El sistema lleva en funcionamiento desde 1992 y permite generar los informes en inglés o francés.

El proceso de conversión de los datos en texto consta de tres etapas: extracción de datos, procesamiento conceptual y procesamiento lingüístico.

El primer paso es extraer datos específicos de la región sobre la que se desee generar el informe, posteriormente el procesado de los datos permite obtener sucesos significativos que son evaluados a su vez por las necesidades del meteorólogo. Esto se realiza a través de un sistema experto que a partir de esos valores sin procesar como entrada, genera una representación conceptual que posteriormente será la base del informe generado.

Las Figuras 2.6 y 2.7 representan las entradas del sistema FoG y las salidas del mismo.

Una complejidad en el sistema es que debe decidir que detalles de la información debe proporcionar. Necesita decidir cuándo puede usar términos temporalmente inexactos como «this evening» o «early Friday evening» y cuando debe usar frases más precisas como «midnight».

### 2.6.3. STOP

STOP Smoking Termination through cOmputerised Personalisation) [Reiter et al. (1999)] es capaz de generar documentos personalizados con el objetivo de dejar de fumar. El sistema presenta dos modos de funcionamiento, por

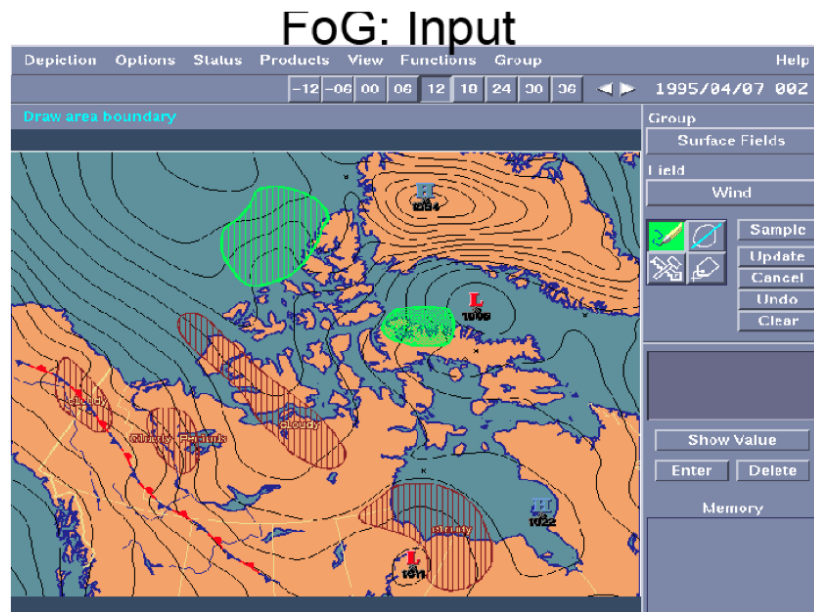


Figura 2.6: Entrada de FOG: Sistema meteorológico sobre Canada

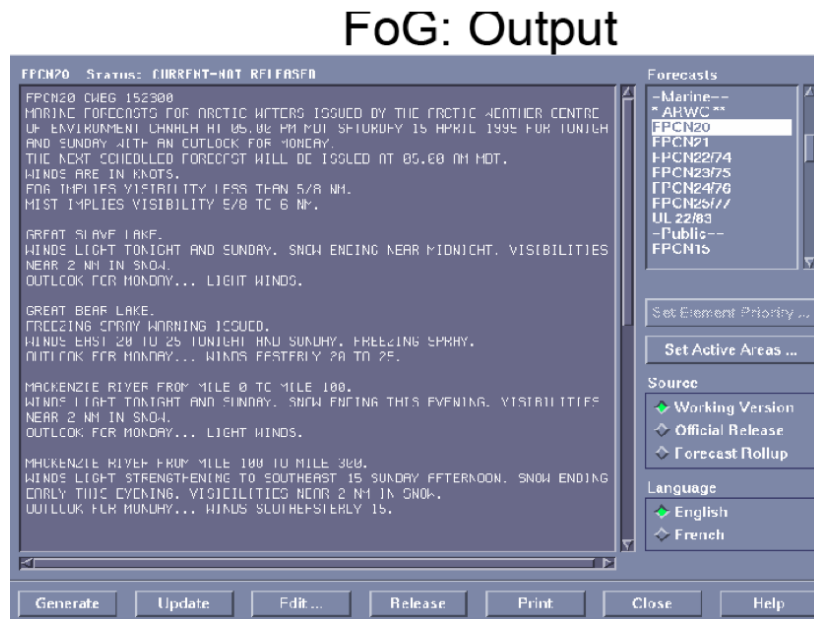


Figura 2.7: Salida de FOG

un lado es capaz de generar la salida como un HTML a partir de un script CGI, por otro, permite crear un fichero RTF tomando como entrada una base de datos creada a través del software Microsoft Access.

Como se ha dicho los documentos generados son personalizados para cada usuario, para ello el sistema utiliza las respuestas que han dado los usuario a ciertos cuestionarios sobre sus hábitos y creencias con realización al tabaco. El sistema trata de generar salidas en las que no se estimule de forma negativa al usuario, es decir, no se le regañe, sino que trata de animar y exaltar los beneficios de dejar de fumar, para ello expresa en sus salidas consejos prácticos que el usuario puede llevar a cabo. En la Figura 2.8 se puede ver la entrada del sistema.

STOP : Input

**SMOKING QUESTIONNAIRE**

Please answer by circling the number which best describes you.

**Q1. How long have you smoked?**  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85  90  95  100

**Q2. How many cigarettes do you smoke per day?**  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50

**Q3. How many cigarettes do you smoke per week?**  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40

**Q4. How many cigarettes do you smoke per month?**  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40

**Q5. How many cigarettes do you smoke per year?**  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40

Figura 2.8: Entrada del sistema STOP

Una de las características más interesantes del sistema STOP es la técnica de adquisición de conocimiento que los desarrolladores usan para interactuar con el experto del dominio. Basadas en técnicas desarrolladas por investigadores en sistemas expertos, se incluye un protocolo de análisis y ordenación. Otra característica llamativa es el uso de elementos tipográficos y multimodales como son distintos tipos de letras, listas detalladas y gráficos embebidos.

El sistema fue desarrollado en la Universidad de Aberdeen y es usado por los servicios de salud británicos, además genera una salida similar a la de la Figura 2.9.

#### 2.6.4. ERMA

ERMA [Clippinger (1974)] es un sistema que tiene su origen en un proceso de investigación sobre las fases necesarias para generar lenguaje natural. El sistema es capaz de generar un monólogo representando una persona que

## STOP: Output

**Dear Ms. Cameron**

Thank you for taking the trouble to return the smoking questionnaire that we sent you. It appears from your answers that although you're not planning to stop smoking in the near future, you would like to stop if it was easy. You think it would be difficult to stop because smoking helps you cope with stress, it is something to do when you are bored, and smoking stops you putting on weight. However, you have reasons to be confident of success if you did try to stop, and there are ways of coping with the difficulties.

Figura 2.9: Salida del sistema STOP

interactúa con un psicoanalista, para ellos los textos generados se intentan aproximar a las transcripciones reales de los pacientes.

ERMA consta de varios módulos que permiten la generación del texto: Calvino (recolección y censura de contenidos), Maquiavelo (organización del texto), Cicerón (realización), Freud (observación de los orígenes de las tareas de generación) y Leibniz (una red de definición de conceptos). Además, cada una de las diferentes partes del programa producen efectos en el texto para generar las expresiones de duda y momentos en blanco, tal y como lo hacen los pacientes.

Otra característica destacable de ERMA es que es capaz de re-planificar un texto anteriormente planificado para expresarlo mejor.

### 2.6.5. MEXICA

MEXICA es un sistema basado en el modelo creativo desarrollado por [Pérez y Pérez (1999)] y fundamentado en Engagement-Reflection (E-R). MEXICA [Pérez y Pérez y Sharples (2001)] es un programa capaz de generar cuentos sobre los mexicas (antiguos habitantes del Valle de México), ver Figura 2.10.

El sistema MEXICA está basado en el modelo (E-R), este modelo contiene ideas de diferentes investigadores y Mike Sharples [Sharples (1999)] recoge intentando definir el proceso creativo que se produce cuando intentamos escribir un texto o un cuento. El proceso recogido por Sharples consiste en un ciclo constante entre dos estados mentales conocidos como Estado-E (Engagement, ensimismamiento) y Estado-R (Reflection, reflexión). Cuando una

personas se encuentra en el Estado-E, se encuentra inmersa en la generación de nuevas ideas, que a su vez generan nuevas ideas por la asociación de las mismas. Este proceso tiene lugar hasta que, o bien somos interrumpidos, o bien no somos capaces de generar nuevas asociaciones y por tanto nuevas ideas. Durante este periodo solo se realiza la generación de nuevas ideas pero no se evalúan ni se obtienen conclusiones sobre la calidad de las mismas. Cuando el Estado-E se ve interrumpido se puede pasar al Estado-R, en este estado se realiza una evaluación del material o ideas generadas, se trata de comprobar que cumplen los requerimientos necesarios de la tarea que en la que no encontramos embarcados (no es lo mismo escribir un cuento para niños que un cuento de terror). También se evalúan otros factores, como puede ser la calidad del material producido o el grado de entretenimiento de las ideas generadas.

Con el desarrollo de MEXICA se trato de conseguir que las historias generadas fueran novedosas, coherentes e interesantes. Para ello el usuario puede definir un gran número de parámetro que producirán historias con diferentes características en función de los parámetros especificados. MEXICA para valorar la novedad de una historia creada, almacena datos de los cuentos creados en una base de conocimientos. Un cuento es interesante si existen cambios en la tensión dramática del cuento.

Para evaluar las historias desarrolladas por MEXICA se diseño un cuestionario que contestaron cincuenta personas de doce diferentes países. En él se comparaban cuatro cuentos generados por MEXICA, dos cuentos generados por otros programas similares, y un cuento hecho por un humano empleando el mismo tipo de lenguaje que el producido por los programas de cómputo. Se pidió a los sujetos que evaluaran en una escala del 1 al 5 la coherencia y fluidez, estructura, contenido, suspenso y calidad de las siete historias. Los resultados colocaron a MEXICA en el primer lugar de las evaluaciones. Estos resultados sugieren que MEXICA es capaz de producir cuentos al menos con la misma calidad que otros programas de cómputo.

Con el desarrollo de MEXICA se ha podido comprobar que el modelo E-R es válido para el estudio del proceso creativo en sistemas de computo. Además es el primer sistema que es capaz de auto-evaluar una historia creada por el mismo en comparación con su base de conocimiento.

## 2.7. Entorno virtual o modelo 3D

Se denomina entorno virtual a una representación gráfica de un modelo del mundo real, esto se consigue a través de un proceso de renderizado en el que se genera una imagen en 2D a partir de una en 3D. Más concretamente, la imagen en 2D se construye a partir de una definición de un modelo (real o abstracto) en 3D.

El origen de este tipo de modelos se encuentra en la simulación militar

*Jaguar\_knight was an inhabitant of the great Tenochtitlan. Princess was an inhabitant of the great Tenochtitlan. From the first day they met, Princess felt a special affection for Jaguar\_knight. Although at the beginning Princess did not want to admit it, Princess fell in love with Jaguar\_knight. Princess respected and admired Artist because Artist's heroic and intrepid behaviour during the last Flowery-war. For long time Jaguar\_knight and Princess had been flirting. Now, openly they accepted the mutual attraction they felt for each other. Jaguar\_knight was an ambitious person and wanted to be rich and powerful. So, Jaguar\_knight kidnapped Artist and went to Chapultepec forest. Jaguar\_knight's plan was to ask for an important amount of cacauatl (cacao beans) and quetzalli (quetzal) feathers to liberate Artist. Princess had ambivalent thoughts towards Jaguar\_knight. On one hand princess had strong feelings towards Jaguar\_knight but on the other hand Princess abominated what Jaguar\_knight did. Suddenly, the day turned into night and after seconds the sun shone again. Princess was scared. The Shaman explained to Princess that Tonatiuh (the divinity representing the sun) was demanding Princess to rescue Artist and punish the criminal. Otherwise Princess's family would die. Early in the Morning Princess went to Chapultepec forest. Princess thoroughly observed Jaguar\_knight. Then, Princess took a dagger, jumped towards Jaguar\_knight and attacked Jaguar\_knight. Jaguar\_knight was shocked by Princess's actions and for some seconds Jaguar\_knight did not know what to do. Suddenly, Princess and Jaguar\_knight were involved in a violent fight. In a fast movement, Jaguar\_knight wounded Princess. An intense haemorrhage arose which weakened Princess. Jaguar\_knight felt panic and ran away. Thus, while Tlahuizcalpantecuhli (the god who affected people's fate with his lance) observed, Princess cut the rope which bound Artist. Finally, Artist was free again! Princess was emotionally affected and was not sure if what Princess did was right. Princess was really confused. The injuries that Princess received were very serious. So, while praying to Miclantecuhli (the lord of the land of the dead) Princess died.*

Figura 2.10: Fragmento generado por MEXICA

y en los simuladores de vuelo, además, ha sufrido un gran impulso gracias a las mejoras software y hardware, ya que el procesamiento necesario para la generación de imágenes es muy costoso computacionalmente hablando, por lo que requiere de máquinas con cierta potencia. Las mejoras software han permitido que las imágenes generadas sean cada vez más próximas al mundo real, el desarrollo de vídeo juegos ha sido uno de los campos que más ha favorecido el crecimiento de esta tecnología.

Existen muchos ejemplos de utilización de entornos virtuales 3D, uno de los más representativos y expandidos es Second Life (SL), sus usuarios, conocidos como residentes", pueden acceder a SL mediante el uso de uno de los múltiples programas de interfaz llamados viewers (visores), los cuales les permiten interactuar entre ellos mediante un avatar. Los residentes pueden así explorar el mundo virtual, interactuar con otros residentes, establecer relaciones sociales, participar en diversas actividades tanto individuales como en grupo y crear y comerciar propiedad virtual y servicios entre ellos.



Figura 2.11: Imagen de la réplica en Second Life del centro de estudios localizado campus universitario de Saint Leo, Florida.



## Capítulo 3

# Planificación, Diseño e Implementación de la Aplicación

En este apartado se va a explicar cómo se ha planificado y diseñado la aplicación base sobre la que se centra el proyecto y los posteriores experimentos. En otros apartados ya se dieron una pinceladas de los pasos que pretenden realizar de forma general dentro del proyecto, en este caso se comentará el desarrollo de la aplicación. Se ha realizado un desarrollado de forma iterativa e incremental, de forma que en cada una de las iteraciones se cumplan los requisitos identificados en la iteración anterior y se reconozcan nuevos requisitos y mejoras para la siguiente iteración.

### 3.1. Primer prototipo

Al inicio de este proyecto se encontraba disponible un modelo 3D del edificio que podría ser utilizado. Por tanto, se decidió hacer uso del mismo y desarrollar una aplicación que hiciera uso del modelo ya disponible. Con el primer prototipo lo que se pretendía inicialmente era desarrollar una aplicación que permitiera cargar el modelo y que un usuario pudiera desplazarse de forma sencilla por el interior del mismo. De esta forma, los tiempos necesarios para realizar el recorrido se verían ampliamente reducidos. Otra ventaja que presenta este primer prototipo es la posibilidad de realizar el experimento con un amplio número de personas de forma simultánea, así es posible evaluar la validez de un conjunto de instrucciones entregándoselas a varias personas para que traten de realizarlas.

El resultado de esta primera iteración es el que se ve en la Figura «3.1». Como se puede observar, este primer prototipo carga el modelo de la facultad y permite navegar a través del mismo haciendo uso del teclado y del ratón.



Figura 3.1: Primer prototipo de la aplicación

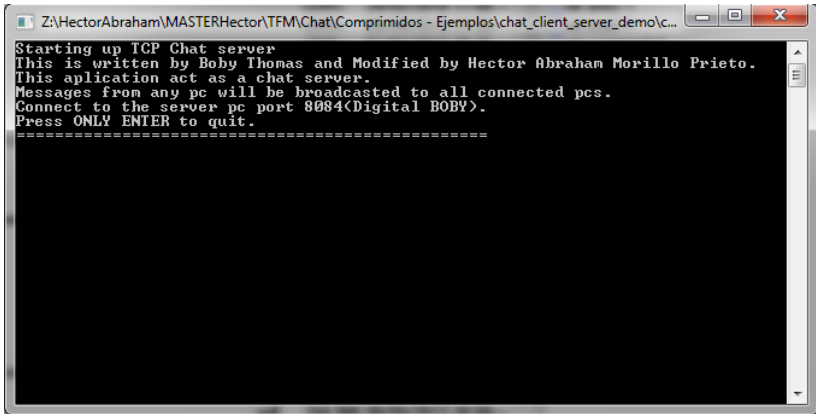
Para poder cargar el modelo inicialmente se intentó reutilizar parte del código de un proyecto anterior. El problema que apareció estaba en la versión del modelo 3D, ya que este estaba preparado para una versión más moderna de OGRE 3D. Por lo tanto, fue necesario rechazar todo ese código y construirlo de nuevo. El primer paso fue descargar el código fuente del motor de renderizado OGRE 3D y compilarlo, después de esto ya se construyó la aplicación encargada de mostrar el modelo y permitir la navegación por el mismo.

En este prototipo inicial se detectaron algunos errores o problemas. Por ejemplo, dentro del interior del edificio existían puntos que no se encontraban adecuadamente iluminados. Esto provocaba una cierta dificultad en la visualización de los elementos que se encontraban en el modelo. Por otro lado, se detectaron nuevas necesidades para la realización del experimento a través de la aplicación: era necesario que el guía pudiera comunicarse con la persona que realiza la ruta de forma cómoda y que a su vez queden registradas todas las indicaciones para su posterior análisis. También era necesario tener un registro de todo los movimientos que realizaba el usuario, y qué es lo que estaba viendo en los momentos de duda.

### 3.2. Segundo prototipo

Para resolver las necesidades en el prototipo anterior, se planteó desarrollar un chat dentro de la propia aplicación. Para ello, se implementó un

servidor al cual se conectaría cada una de las instancias de la aplicación. En el servidor se encontraría el guía, dando las instrucciones a todos aquellos que se encontraran dentro de la aplicación. Por otro lado también fue necesario implementar dentro de la aplicación (Primer prototipo), la capacidad de conectarse con el servidor desarrollado, visualizar los mensajes recibidos y permitir al usuario escribir y enviar sus propios mensajes.



```
Z:\HectorAbraham\MASTERHector\TFM\Chat\Comprimidos - Ejemplos\chat_client_server_demo\c...
Starting up TCP Chat server
This is written by Boby Thomas and Modified by Hector Abraham Morillo Prieto.
This application act as a chat server.
Messages from any pc will be broadcasted to all connected pcs.
Connect to the server pc port 8084(Digital BOBY).
Press ONLY ENTER to quit.
=====
```

Figura 3.2: Servidor chat desarrollado en la segunda iteración

Como se puede ver en la Figura «3.2», el servidor de chat desarrollado se encuentra en una línea de comandos. El guía debe escribir en esa terminal todas las indicaciones que desee enviar al usuario que se encuentra navegando por el modelo. Este verá dichos mensajes a través de la ventana habilitada a tal efecto. Como se comentaba anteriormente, ha sido necesario dotar al primer prototipo de capacidad para la conexión con el servidor, así como posibilidad para visualizar los mensajes recibidos y aquellos que se desea enviar. En la figura siguiente se muestra el resultado de la implementación del chat en la aplicación «3.3»;

Como se puede observar, se trata de una visualización sencilla pero suficiente para realizar la comunicación entre guía y guiado.

La otra cuestión de gran importancia detectada en la iteración anterior es recoger todos los movimientos e información importante que pueda ser utilizada a la hora de obtener resultados del experimento. Para ello se decidió almacenar toda la ruta realizada por el usuario, lo que incluye posiciones y dirección sobre la que se encuentra mirando dentro del modelo. Es fundamental almacenar también toda la comunicación realizada a través del chat, ya que esto facilitará analizar las forma de expresión utilizadas por cada uno de los intervinientes. Para mejorar la información obtenida, se decide realizar capturas de la imagen que está visualizando el usuario que se encuentra frente al modelo cada vez que escribe un mensaje a través del chat.

En la siguiente figura se puede ver un ejemplo del log generado «3.4»:

Al finalizar este prototipo se detectó un problema importante a la hora

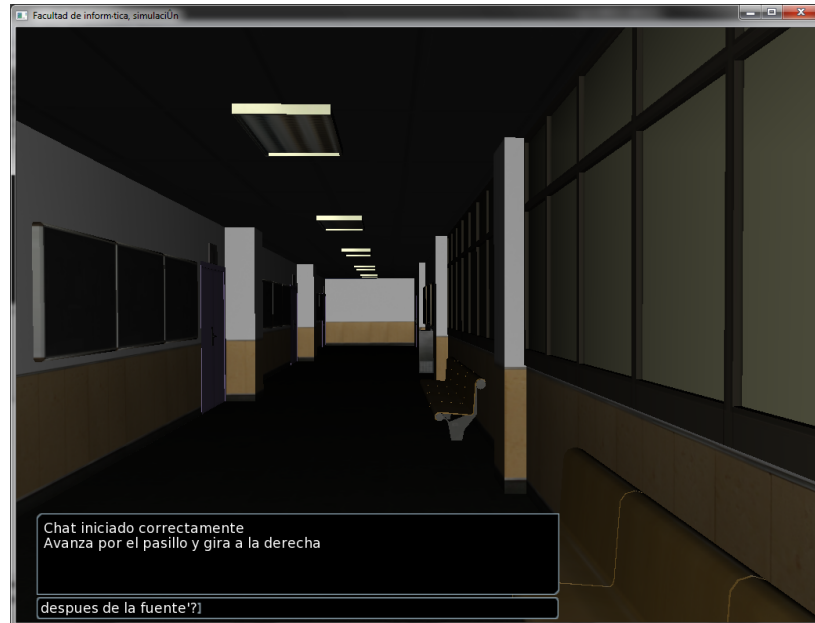


Figura 3.3: Implementación del chat en el segundo prototipo

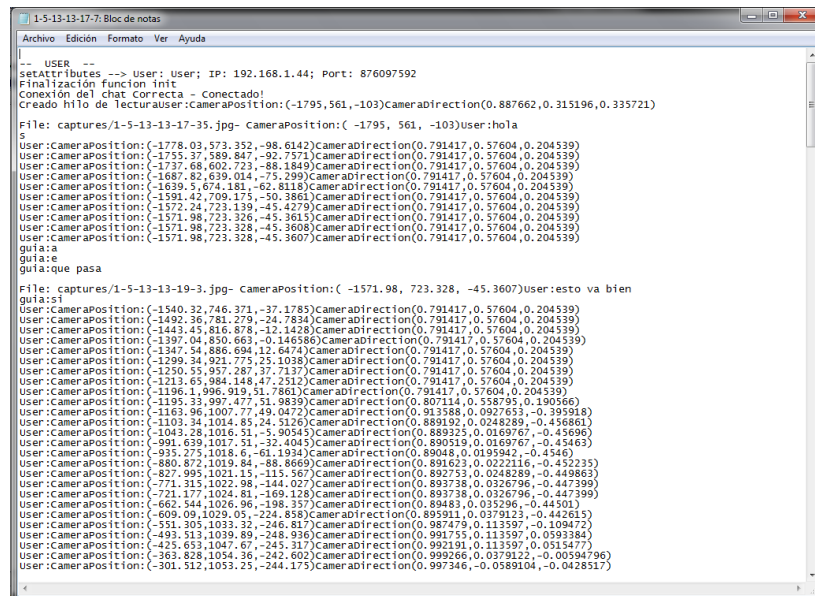


Figura 3.4: Log generado en una ejecución de la aplicación

de realizar el guiado: el guía que se encuentra en la parte servidora del chat no tiene posibilidad de visualizar dónde se encuentra el guiado dentro del modelo por lo que esto dificulta la experiencia de usuario. Una solución sería que a la hora de realizar el experimento ambas personas se situaran

en máquinas próximas, de forma que el guía pudiera ver la pantalla del guiado. El problema que presenta esta solución es que puede interferir en los resultados del experimento ya que involuntariamente el guía podría dar alguna indicación hablada o gestual que no quedaría registrada.

Se llegó a la conclusión de que era necesario que el guía tuviera la posibilidad de ver a distancia (por ejemplo, situando a guía y a guiado en diferentes salas) lo mismo que el guiado para poder realizar las indicaciones.

Por otro lado, sería interesante poder comparar el recorrido realizado sobre nuestro modelo con los mapas utilizados en el proyecto «Nuñez González (2011)», las referencias de coordenadas utilizadas en ambos proyectos son diferentes por lo que es necesario encontrar una función de traducción de las coordenadas entre sistemas. Disponer de esta función de traducción puede ser útil en los futuros análisis.

### 3.3. Tercer prototipo

La idea que surgió para solucionar el principal inconveniente al segundo prototipo de la aplicación fue ampliar la implementación de la aplicación para permitir la ejecución en dos modos diferentes, por un lado el de la persona que necesita ser guiada, que ya estaba implementado, y por otro una nueva funcionalidad que permita arrancar la aplicación en modo guía. Este modo de ejecución debe ser similar al modo de guiado a diferencia de que en este caso el usuario no debe ser capaz de mover la cámara por el modelo, sino que simplemente debe ver en su pantalla lo mismo que está viendo la persona guiada.

El diseño de esta nueva funcionalidad se basa en el chat ya implementado. Se mantiene el servidor chat de la misma forma que estaba implementado y así se pueden conectar diferentes instancias de la aplicación al mismo. Las modificaciones se han realizado en la aplicación que carga el modelo. Cuando la aplicación sea lanzada para navegar a través de la misma, es decir, para ser la persona guiada, se enviarán a través de mensajes «ocultos», la posición y dirección de la cámara en la que se encuentra en cada momento. El servidor del chat se encargará de reenviar todos los mensajes, ya que él no diferencia si es un mensaje escrito por el usuario o uno de los mensajes «ocultos» con las coordenadas actuales. Se denominan mensajes «ocultos» porque son totalmente invisibles para los usuarios, aunque se utilice el servicio de chat para ser enviados. La nueva parte de la aplicación desarrollada se encarga de cargar el modelo y actualizar la posición que se está visualizando cada vez que recibe un mensaje con las coordenadas. De esta manera también se permite que existan varios guías para un único usuario guiado, ya que simplemente deben conectarse al mismo servidor. El guía también dispone de la posibilidad de comunicación a través del chat, de forma que ahora el guía no interactúa directamente con el servidor, sino que crea un cliente

similar al guiado pero sin posibilidad de mover la cámara a través del modelo.

En esta iteración de la aplicación también se han calculado las ecuaciones de traducción de las coordenadas entre nuestro modelo y el del proyecto «Nuñez González (2011)», que utilizaba un mapa en 2D como la Figura 3.5:

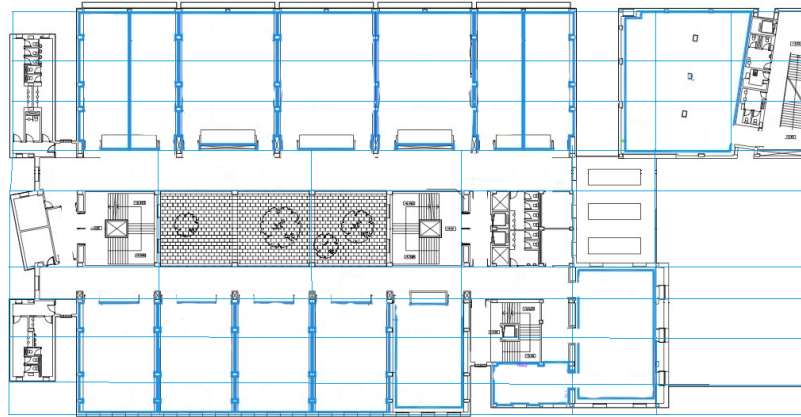


Figura 3.5: Mapa 2D utilizado en el proyecto de Elena Núñez González

Como en ese proyecto se realiza el cálculo de trayectorias dentro de la facultad y disponemos de una función de traducción podríamos simular uno de los recorridos generados en nuestro proyecto para analizar los datos recogidos.

Una utilidad que también se añadió en esta versión del prototipo es la de permitir a un usuario guiado situarse en unas coordenadas concretas. Esto tiene gran utilidad a la hora de desarrollar los experimentos, ya que puede permitir realizar diferentes recorridos previamente preparados, situando al usuario en la posición inicial deseado de una forma muy sencilla. Para introducir esta orden a la aplicación se hace uso del cuadro de texto habilitado para el chat, donde a través de una orden similar a la siguiente: «PUTMEIN-xxx.x-yyy.y-zzz.zz», se traslada la cámara utilizada por el usuario a la posición indicada. Esto facilita en gran medida la realización de diferentes rutas con los mismos usuarios, ya que reduce el tiempo necesario para iniciar otra ruta ya que el usuario no se tiene que desplazar manualmente a través del modelo y además realiza mucho más sencillo y eficaz.

## 3.4. Técnicas y herramientas

### 3.4.1. OGRE 3D

OGRE es un flexible motor tridimensional orientado a escenas y escrito en C++. Está diseñado para hacer más fácil e intuitivo para los desarrolladores la programación de aplicaciones que utilizan aceleración gráfica 3D por hardware. La biblioteca de clases de OGRE abstrae al programador de



Figura 3.6: Logotipo Ogre 3D

las clases y funciones de las bibliotecas subyacentes sobre las que funciona: Direct3D y OpenGL. La interfaz basada en objetos de OGRE simplifica al desarrollador la creación de cámaras, la carga de mallas de objetos y muchas otras de las acciones más comunes que se realizan en una aplicación de este tipo.

La principal ventaja de OGRE, y de cualquier otra interfaz gráfica, es la minimización del esfuerzo a la hora de renderizar escenas 3D, al estar la implementación 3D abstraída por el motor. De este modo, el programador no tiene que ocuparse de operaciones como el culling jerárquico, las transparencias o el manejo del estado del renderizado. Además, su diseño orientado a objetos facilita enormemente el manejo de escenas complejas.

Otra de sus grandes ventajas es su sistema de materiales y sombreados. Estos son cargados de ficheros de texto independientes del código, lo cual simplifica el código enormemente desacoplándolo y haciendo de la modificación de materiales y shaders algo mucho más cómodo y simple.

OGRE es completamente gratuito y está disponible bajo la licencia MIT. El código fuente está a disposición de cualquiera y todo el mundo puede modificarlo, dejando a su elección si publicar dichas mejoras en su comunidad o no. La gratuidad de OGRE, hace que exista una comunidad de desarrolladores bastante amplia que se dedica a crear bibliotecas, escribir tutoriales y ayudar a sus usuarios por medio de Wikis y foros. Adicionalmente, se encuentra disponible una amplia documentación para todas sus clases. Todas estas razones propician que numerosos videojuegos y aplicaciones, tanto amateur como profesionales, hayan sido realizados utilizando OGRE.

En este proyecto se utiliza Ogre 3D para cargar el modelo en 3D de la facultad de informática que se encuentra disponible.

### 3.4.2. CEGUI

Ogre dispone de una librería para desarrollar la GUI de las aplicaciones muy versátil y potente llamada CEGUI.

CEGUI es una biblioteca libre para la creación de ventanas y widgets para los motores donde la funcionalidad no está disponible de forma nativa. La biblioteca está orientada a objetos, escrito en C++, y está dirigida a los desarrolladores de aplicaciones de manera que no deban perder gran cantidad de tiempo en la construcción de los subsistemas de interfaz gráfica de usuario.

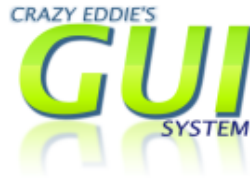


Figura 3.7: Logotipo librería CEGUI

En nuestro caso se ha utilizado las funcionalidad de CEGUI para implementar la parte visual de la funcionalidad del chat dentro de la aplicación encargada de cargar el modelo. Más concretamente la interfaz rectangular que permite al usuario escribir un mensaje y ver todos aquellos recibe.

### 3.4.3. Sockets - Chat

Un socket queda definido por la dirección IP de la máquina, el puerto en el que escucha, y el protocolo que utiliza. Los sockets se clasifican en sockets de flujo (`socket.SOCKSTREAM`) o sockets de datagramas (`socket.SOCKDGRAM`) dependiendo de si el servicio utiliza TPC, que es orientado a conexión y fiable, o UDP, respectivamente. Los sockets también se puede n clasificar según la familia. Tenemos sockets UNIX (`socket.AFUNIX`) que se crearon antes de la concepción de las redes y se basan en ficheros, sockets `socket.AFINET`, `socket socket.AFINET6` para IPv6, etc.

En nuestro caso de ha implementado un servidor a través de sockets que se encuentra escuchando continuamente en un puerto de la máquina en la que se lance. Todos los mensajes que sean recibidos se reenviaran al resto de conexiones que se hayan establecido con ese servidor. Es decir, el servidor realiza un broadcast del mensaje recibido hacia todos los canales a través de los que se ha abierto un socket TPC. Como es lógico se puede lanzar la instancia del servidor en cualquier máquina siempre que tenga posibilidad de comunicarse con el resto de maquinas donde se va a lanzar la aplicación.

Este servicio basado en sockets se utiliza con dos objetivos fundamentales:

- Mensajería chat: Permite la comunicación entre usuarios de la aplicación, es uno de los pilares de la aplicación ya que permite almacenar para posteriormente analizar las expresiones utilizadas.
- Comunicación entre instancias de la aplicación: Las diferentes instancias de la aplicación necesitan comunicarse para poder enviar o recibir la posición del usuario «guiado». Es decir, la instancia del usuario «guiado» será la encarga de enviar a través de los socket de comunicación abiertos con el servidor la posición y dirección en la que se encuentra, el servidor reenviará estos datos a todos los «guías» que se encuentren conectados para que las instancias de la aplicación puedan

---

colocar la cámara en la misma posición y dirección de forma que ese usuario pueda tener en su pantalla exactamente lo mismo que ve el usuario «guiado».



## Capítulo 4

# Diseño de los experimentos

En este capítulo se pretende describir el diseño de los experimentos, así como lo que se trata de conseguir con ellos. Se tratará de detallar los objetivos que se desean conseguir y como se ha planteado la realización de los mismos, posteriormente comentaremos el desarrollo de un primer estudio piloto para encontrar debilidades en el diseño realizado y corregirlas antes de realizar experimentos de cierta entidad.

### 4.1. Introducción

Como ya ha sido comentado en apartados anteriores, la clave de este proyecto reside en crear un medio que permita recopilar gran cantidad de información sobre cómo las personas dan indicaciones a otras personas en función del contexto en el que se encuentren.

El estudio realizado se centra fundamentalmente en la monitorización y análisis de instrucciones en interiores. Por ello, la aplicación desarrollada como entorno de experimentación recrea el interior de un edificio. Esto se debe a que no se utilizan los mismos recursos del lenguaje para dar instrucciones en recintos abiertos que en recintos cerrados. Además, tampoco se dan las instrucciones de la misma manera cuando la persona que guía sabe en todo momento dónde se encuentra la persona guiada que cuando el guía debe de una serie de instrucciones, todas en conjunto, para que luego el guiado, utilizando dichas instrucciones, llegue al objetivo marcado inicialmente.

Con los primeros experimentos que se han realizado se pretende verificar el correcto funcionamiento de la aplicación, comprobar que la aplicación se ejecuta correctamente en diferentes equipos y que la comunicación a través de la red es correcta. Hay que tener en cuenta se envía gran cantidad de mensajería para la sincronización de la aplicación de la persona que está siendo guiada, que es quien realiza el movimiento dentro del modelo, y la aplicación del guía, donde se tiene que ver reflejado todos los movimientos que se realicen. Para esta sincronización, la aplicación en modo guiado envía

información de su posición cada cuarenta milisegundos, por lo que en función del número de personas que se encuentren realizando el experimento dentro de una misma red, la cantidad de tráfico que se genere puede ser elevado. Se desconoce por tanto cómo se va a comportar la red, aunque se espera que no exista ningún tipo de problema debido a la capacidad de las redes actuales.

También se pretende comprobar si la aplicación desarrollada es intuitiva para los usuarios y fácil de usar. Esto puede tener cierta relevancia en los resultados que se puedan obtener a través de este entorno de experimentación. Si los usuarios no se encuentran cómodos o encuentran dificultades a la hora de utilizar la herramienta, posiblemente su interés o motivación por realizar el experimento desaparecerá, y por tanto pondrán menor atención o predisposición que si sus indicaciones se encontraran en un contexto del mundo real. Por otro lado, el hecho de que encuentren dificultades hará que sea necesario darles indicaciones sobre su utilización o sobre las acciones que tienen que realizar. Esto es contrario a uno de los puntos planteados en la introducción de este documento, donde se dice que se busca la mayor naturalidad y espontaneidad posible para que los resultados se ajusten en mayor medida a lo que sucedería en el mundo real.

No sólo es interesante verificar el correcto funcionamiento de la aplicación; ya se ha comentado que el objetivo central es facilitar la adquisición de datos. Por tanto, es necesario que a través de estos experimentos iniciales se demuestre que la información generada y almacenada por la aplicación es suficiente para un análisis posterior en el que se pueda identificar los elementos que utilizan las personas para dar indicaciones, como por ejemplo si utilizan el color de los elementos, la forma o alguna otra característica, como hacen referencia a las distancias o como diferencian entre elementos similares. Es decir, es importante no sólo conocer que se utiliza para expresar las indicaciones, sino también, cuando se utiliza cada una de esas posibilidades. En este caso hay que comprobar que el log generado permite reproducir toda la conversación mantenida entre las personas involucradas en el experimento y no sólo eso, sino también determinar el recorrido realizado de forma que pueda ser regenerado sobre el modelo. Por este motivo, como se puede ver en el «Apéndice C», se almacenan las diferentes posiciones y direcciones de la cámara por las que se va moviendo el usuario a lo largo del recorrido. Con esta información se podría, si fuera necesario, reproducir el recorrido sobre el modelo 3D, incluso reproduciendo las conversaciones que se mantuvieron en cada uno de los momentos, ya que esa información queda registrada en el mismo log. Un detalle importante son las capturas de pantalla realizadas en cada máquina cuando un usuario envía un mensaje. Se pueden aprovechar estos experimentos para comprobar el soporte que dan dichas imágenes para entender los logs generados.

## 4.2. Descripción

En esta sección se va a describir cómo se ha planteado la realización de los experimentos y las condiciones que se plantean como óptimas para el correcto desarrollo de los mismos. Durante estos experimentos se tratará de conseguir lo explicado anteriormente. Además, también servirá como una primera aproximación al uso general que podría tener la aplicación.

Para el desarrollo de los experimentos se decide plantear una serie de rutas dentro de la propia facultad. Cada una de estas rutas tiene un punto de origen y un punto de destino claramente identificados. Las rutas se han planteado de forma que todas ellas tengan algún tipo de interés concreto, es decir, necesiten pasar por sitios complicados como pueden ser espacios abierto con varias salidas, objetos similares entre los que el guía deba realizar una diferenciación etc. Se trata de buscar tanto situaciones comunes como otras que requieran de mayor tipo de explicaciones o en las que se deban utilizar otros recursos del lenguaje menos frecuentes.

Las rutas planteadas, junto con el interés esperado de la misma, pueden ser consultadas en el «Apéndice B». En ese documento también se puede observar las coordenadas de los puntos de inicio de cada una. Esta información es muy útil a la hora de realizar el experimento, ya que los usuarios tan solo deben introducir en la línea de entrada del chat la orden que permite trasladar la posición actual a la coordenadas introducidas. Lo que se consigue así es que todos los usuarios comiencen una misma ruta desde el mismo punto y poder así comparar las indicaciones dadas por diferentes usuarios.

Para realizar el experimento se deben tener en cuentas algunas consideraciones que favorecerían mejores resultados:

- Tratar de dar el menor número de indicaciones posibles tanto a la persona guiada como al guiado, de forma que no se vean influenciados por aquellos que estén llevando el experimento a cabo. Para que puedan conocer el funcionamiento de la aplicación se ha preparado un breve manual de usuario [Apéndice A] que se centra únicamente en cómo iniciar la aplicación y funcionamiento interno de la misma, chat, movimiento de la posición etc.
- Dentro de las posibilidades del lugar donde se vaya a realizar el experimento sería recomendable que se instalaran dos salas aisladas pero con capacidad de comunicación por red. En una de esas salas se colocaría a aquellos voluntarios que inicialmente fueran a actuar como guías y en la otra aquellos que vayan a ser guiados. Una vez realizadas las rutas se invertirán los papeles de forma que los guías pasen a ser guiados y viceversa. Con esta separación se consigue que no puedan tener comunicación directa y por tanto toda la comunicación se deba realizar a través del chat.

- En caso de que el punto anterior no sea posible y todas las personas participantes en el experimento tengan que encontrarse en la misma sala, se debería, si es posible, organizar a guías y guiados de forma que no conozcan ni sepan donde está situada la persona con la que se están comunicando.
- Una vez finalizado el experimento se deben recopilar los logs de ambos usuarios, aunque a través de uno de los logs generados se puede reconstruir tanto el recorrido realizado como la conversación mantenida. Las capturas realizadas cada vez que se escribe a través del log se almacenan de forma local en cada una de las máquinas que se encuentren ejecutando la aplicación y no se transmiten a través de la red a la otra máquina.

Los pasos genéricos para realizar un experimento serían los siguientes (habría que añadir, si fuera necesario, puntos concretos que requieran experimentos con objetivos específicos):

1. Preparar la sala/salas en las que se va a realizar el experimento, es necesario copiar en cada uno de los equipos el ejecutable del entorno de experimentación.
2. Determinar qué equipos serán los encargados de ejecutar el servidor chat. Estos equipos pueden ser utilizados también para ejecutar la aplicación.
3. Coordinar a los voluntarios disponibles de forma que al menos exista un guía para cada persona guiada (si se desea es posible que se conecten varios guías).
4. Una vez que todos los usuarios saben el papel que van a desempeñar y se encuentran en su equipo, se les entregará el manual de usuario «Apéndice A» y se les indicará la dirección ip donde se encuentra el servidor al que se tienen que conectar. En ese momento pueden arrancar la aplicación en el modo de ejecución que les corresponda.
5. Si es la primera vez que los usuarios ejecutan la aplicación, se les dará un cierto tiempo para que comprueben el funcionamiento de la misma y puedan adaptarse a sus controles.
6. Se entregará a los guías el documento con las rutas propuestas «Apéndice B». En ese momento los usuario pueden comenzar a realizar el experimento.
7. Una vez finalizado, se recogerán de cada equipo los ficheros de log y las capturas generadas.

### 4.3. Primer estudio piloto

Antes de realizar un experimento de cierta entidad se planteó realizar un estudio piloto en el que participaran un conjunto reducido de voluntarios con el fin de detectar aquellas incidencias o fallos que puedan tener lugar en el planteamiento de los experimentos. Se trata de evitar que esos errores se produzcan directamente en los experimentos que tengan como objetivo recopilar información, y que debido a esos fallos la calidad del resultado se vea disminuida.

El estudio piloto fue propuesto a un grupo de alumnos el día 28 de Mayo de 2013. Aquellos que quisieron pudieron participar de forma voluntaria. Con aquellos alumnos que aceptaron realizar la prueba se realizaron una serie de pasos similares a los propuestos en el apartado «4.2» aunque con algunas diferencias. El estudio piloto ha sido realizado en la Facultad de Informática de la Universidad Complutense de Madrid, donde fue imposible conseguir dos aulas de informática que tuvieran comunicación de red entre ellas. Por tanto, el estudio tuvo que ser realizado en único aula.

Durante el inicio del experimento, a algunos de los participantes les surgieron dudas sobre la inicialización de la aplicación y el modo en el que debían de arrancarlo. Por ello, el manual de usuario fue corregido. Además se detectó uno de los inconvenientes que ya se planteaban en la descripción de los experimentos: al estar situados en un mismo aula y conocer la persona con la que están interaccionando, existen ocasiones en las que de forma involuntaria no se utiliza el chat para dar las indicaciones sino que lo hacen directamente de forma hablada.

Respecto del funcionamiento de la aplicación desarrollada, el estudio piloto confirmó su correcto funcionamiento, tanto en la carga del modelo como en la comunicación a través de la red entre las diferentes instancias de la aplicación y en la información almacenada en los logs y a través de las capturas.

La duración final del experimento fue de aproximadamente un hora y el resultado obtenido fue del agrado tanto del autor del proyecto como de los directores. Aunque el número de participantes no fue muy elevado se pudo comprobar que el planteamiento realizado para el desarrollo de los experimentos puede ser correcto. Además la información generada por el proyecto (logs y capturas) permite deducir algunas características frecuentemente utilizadas dentro del lenguaje, aunque debido al bajo número de participantes no se puede afirmar nada, se puede intuir ligeramente los resultados que se obtendrían con un experimento similar y un tamaño muestral mucho mayor. Por ejemplo, analizando los logs generados se puede ver como todos los guías utilizaron algunas características de un objeto para diferenciarlo entre otros similares que podían causar confusión, «Guía 47:atraviesa la puerta de las barras», se utiliza una característica de la puerta para diferenciarlo entre

otros posibles puertas que puedan causar confusión.

El objetivo perseguido con este estudio piloto por parte de los responsables del proyecto, era el de demostrar la validez del entorno de experimentación desarrollado para el estudio de la generación de instrucciones a través del lenguaje natural. Esto fue conseguido aunque también permitió detectar ciertas mejoras que aumentarían las prestaciones o mejorarían las que ya ofrece el entorno actual de experimentación, esto se verá en la sección «6.2».

## Capítulo 5

# Resultados

En este capítulo se va a comentar brevemente el resultado final de los desarrollos software realizados. Como existe un capítulo exclusivamente dedicado a la evolución y la captura de nuevos requisitos, solo será comentado el resultado final.

La aplicación final implementada hasta el momento de redacción de esta memoria pretende ser una solución práctica para realizar experimentos relacionados con la generación de lenguaje natural. Como ya se ha comentado, para la generación de lenguaje natural es necesario conocer cómo las personas se expresan y esa es una de las mayores dificultades que presenta este área de estudio. A través de la herramienta diseñada e implementada se pretende facilitar el proceso de adquisición de datos.

Como se puede ver en la Figura 5.1 el inicio de la aplicación es idéntico tanto para el modo de funcionamiento de guía como de guiado. Inicialmente se planteó generar dos fichero ejecutables diferentes cada uno de los cuales que incorporara una de la funcionalidades. Pero para simplificar la ejecución de cara al usuario y con objeto de reutilizar la mayor parte del código posible se decidió implementar ambos modos de funcionamiento dentro del mismo ejecutable. En la Figura 5.1 se observa cómo cada usuario puede personalizar el nombre que será mostrado en los mensajes que se envíen a través del chat. Además se debe introducir la dirección ip en la que se encuentra el servidor y el puerto sobre el que escucha.

Una vez arrancada la aplicación existen muy pocas diferencias entre la funcionalidad de guía (figura 5.2) y la de guiado (figura 5.3). A simple vista no existe diferencia alguna ya que ambos visualizan en la pantalla el modelo 3D previamente cargado. El punto que las diferencia se encuentra en su funcionamiento: el guiado tiene posibilidad de desplazarse por el modelo mientras que el guía solo podrá visualizar lo mismo que este viendo en cada momento el guiado.

Como se observa en la parte inferior izquierda de las captura de la aplicación ambos modos de funcionamiento se conectan a través de un servicio de

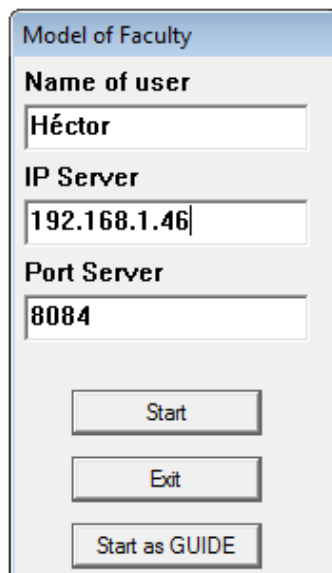


Figura 5.1: Ventana de selección para el inicio de la aplicación

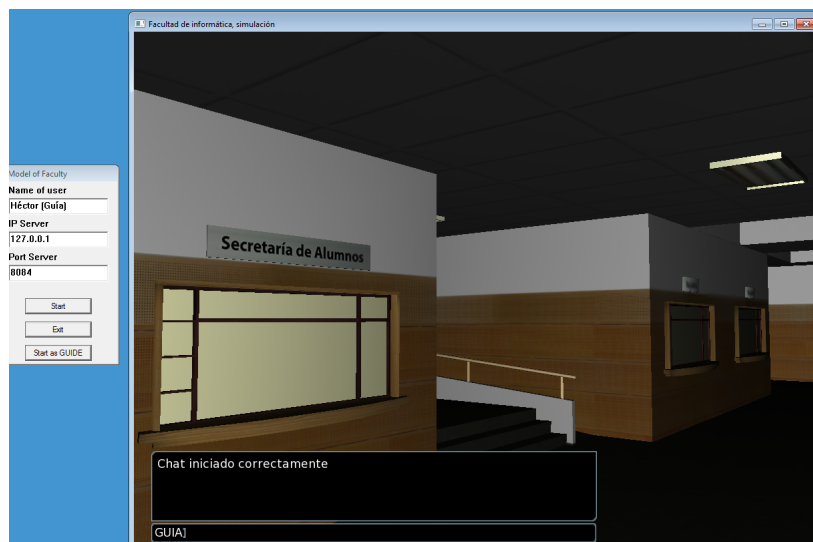


Figura 5.2: Aplicación ejecutando en modo guía

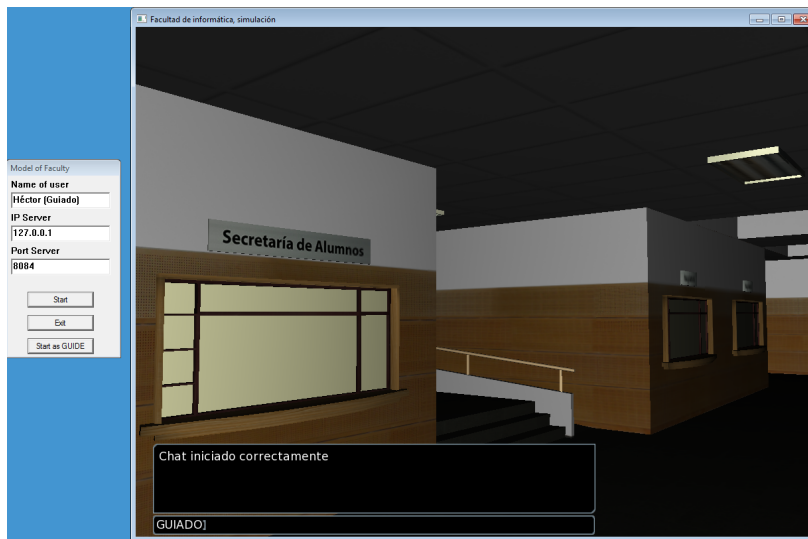


Figura 5.3: Aplicación ejecutando en modo guiado

chat, en la figura 5.4 se observa el servido de chat ejecutándose, básicamente se trata de una aplicación que permite escuchar en un determinado puerto y a través de socket los mensajes que recibe y reenviarlo al resto de socket que se encuentran abierto con la aplicación.

Cada vez que cualquier usuario de la aplicación escriba un mensaje a través del chat la aplicación automáticamente realiza una captura de lo que se está visualizando en ese momento concreto dentro del modelo, la capturas son similares a las de la Figura 5.5.

Por último hay que comentar la gran cantidad de información que genera la aplicación. Para ello, se puede observar en el «Apéndice C» cómo es la estructura de los logs y las relaciones con las capturas.

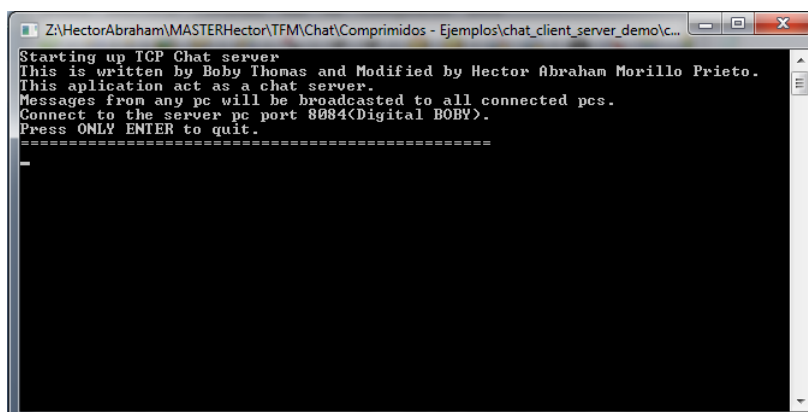


Figura 5.4: Servidor de chat



Figura 5.5: Captura generada por la aplicación

## 5.1. Resultado del experimento piloto

En esta sección vamos a comentar brevemente el resultado del experimento piloto así como que información se extrajo del mismo. En el apartado 4.3 ya se ha tratado sobre cómo fue el desarrollo de esta primera aproximación. El resultado fue muy satisfactorio ya que después de analizar la información generada por el entorno, se comprobó que esta era muy completa y que permitía intuir algunas formas de expresión utilizadas por los guías, a pesar de la escasa dimensión de la prueba.

Como muestra, a continuación se van a comentar algunas expresiones detectadas dentro del experimento:

- Una forma de expresión que aparece en varias ocasiones a lo largo de los logs generados es la siguiente: «Ana:la segunda puerta a tu derecha es el despacho», cuando existen varios elementos similares en una zona del modelo, en este caso las puertas, los guías tienden a identificar el lado en el que se encuentra así como la posición que ocupa entre los elementos similares. Es decir, cuantos elementos tiene que recorrer el guiado en el lado indicado hasta encontrarse con el que realmente le interesa.
- Una forma de expresión que resultó interesante fue la siguiente: «Ana:gira a la izquierda 90°», en este caso el guía hace uso de una dirección y un ángulo para indicar al guiado la dirección que debe de tomar. Esto aunque no se ha visto utilizado en más ocasiones durante el experimento piloto por otros guías, si que fue correctamente entendido por

la persona guiada, siguiendo la ruta deseada por el guía. Por tanto, podría tener interés ver si en otros experimentos se utilizan este tipo de expresiones y si son correctamente aplicadas por los guiados.

- A la hora de dar la indicación al usuario guiado de subir o bajar un número de plantas dentro del edificio, se han detectado dos formas diferentes que llevan a la misma ejecución por parte del usuario, una de la expresiones se basa en indicar la acción, subir o bajar, y el número de plantas, por ejemplo: «Roberto:sube dos pisos». La otra expresión que algunos de los guías han utilizado, se centra únicamente en la planta de destino que tienen que alcanzar, por ejemplo: «Victor:tienes que bajar hasta el bajo». Se ha comprobado a través de las capturas generadas por la aplicación y el resto de la comunicación realizada que ambas instrucciones llevaron al guiado al destino correcto.
- Se ha detectado en varios de los logs generados que algunos guías se basaban en elementos únicos del entorno para indicar el objetivo que tenía que alcanzar el usuario. El ejemplo más claro encontrado se ha dado cuando el usuario se encontraba al inicio de un pasillo y tenía que llegar hasta una de las salas que se encontraban en él, el guía utilizó la expresión: «Raul:pone laboratorio7». Aprovechando los elementos del entorno, en este caso los carteles de las puertas que lógicamente son únicos, consiguió indicar el destino al usuario guiado. Por tanto se podría pensar que estas expresiones tienen gran efectividad.

Este pequeño experimento permite comenzar a identificar datos importantes que pueden ser confirmados con experimentos de mayor amplitud. Si parece claro que la herramienta permite detectar formas de expresión comunes y aquellas que son o no fácilmente identificadas por los guiados.



## Capítulo 6

# Conclusiones y trabajo futuro

En este apartado final, vamos a comentar una serie de conclusiones obtenidas después de la finalización del proyecto, se tratará de realizar una valoración de los resultados obtenidos y se identificarán las partes que pueden ser ampliadas o mejoradas de cara a futuros desarrollos.

### 6.1. Conclusiones

Los objetivos iniciales del proyecto fueron planteados después de detectar el problema existente para realizar experimentos a través de los que obtener datos sobre las formas de expresión utilizadas por las personas, concretamente, la adquisición de datos sobre el tipo de instrucciones utilizadas a la hora de dar indicaciones. Para intentar mejorar la situación en esta rama de estudio se proponen una serie de objetivos tanto tecnológicos como científicos que tiene que cumplir la solución implementada [1.2].

Después de realizar el primer estudio piloto se han obtenido una serie de conclusiones sobre los diferentes objetivos planteados inicialmente, además se ha recogido cierta información que permitirá, como se comenta más adelante, sacar ciertas ideas sobre las tendencias que se siguen a la hora de dar indicaciones.

Se ha comprobado la robustez y el correcto funcionamiento del entorno de experimentación desarrollado, la gran mayoría de los usuarios que participaron en el estudio piloto no necesitó de mayor ayuda que un breve manual para ejecutar la aplicación, lo que verifica su facilidad de uso. Además no se detectó ningún error de funcionamiento durante el experimento y las comunicaciones por red no provocaron ningún inconveniente.

La generación de los logs se produjo de forma correcta, todas las pruebas realizadas durante el experimento generaron el log registrando todas las acciones realizadas. Se comprobaron los log para verificar que se habían almacenado todas las posiciones que había recorrido el usuario de forma que sea posible reconstruir el recorrido realizado junto con las indicaciones da-

das. También se detectó un pequeño inconveniente que será propuesto como trabajo futuro, cada entorno de ejecución genera los logs de forma local lo que hace más difícil la recogida de los mismos.

El sistema de ayuda visual para el análisis de los logs fue programado para que realizara una captura de la parte visualizada en el modelo cada vez que se hacía uso del chat del entorno, de esta forma se puede analizar en que punto se encontraba el guiado cuando recibió una nueva instrucción o bien el momento en el que no sabía continuar. Este sistema junto la generación de logs facilita el análisis de los resultados, ambos han sido utilizados para obtener los puntos del apartado 5.1, lo que demuestra la validez de ambos sistemas para obtener conclusiones.

Respecto de los objetivos científicos que se debían conseguir a través del entorno de experimentación, los primeros logs recogidos dan una ligera idea sobre algunas expresiones que son frecuentemente utilizadas para dar indicaciones, además, también se puede hacer un pequeño estudio sobre la efectividad de las expresiones utilizadas. Aunque no se dispone de una gran cantidad de datos recogidos ya que en el experimento participaron un número reducido de personas, se llegó a la conclusión de que el sistema puede ser de gran utilidad para los estudios previos a proyectos relacionados con la generación de indicaciones a través del lenguaje natural, ya que si los resultados obtenidos con una muestra pequeña permiten sacar ciertas ideas, se espera que un experimento de mayor envergadura pueda conseguir conclusiones mucho más rotundas.

Por otro lado, se han sacado conclusiones respecto de como tiene que realizarse el diseño de los experimentos, se han verificado las pautas dadas en la sección 4.2 para el diseño y la realización de los experimentos, aunque claro está, depende de los objetivos específicos del experimento que se vaya a desarrollar.

Este entorno de experimentación facilita la adquisición de grandes cantidades de datos y permite realizar experimentos de forma simultanea con un número indefinido de personas (limitado por el número de equipos disponibles). Además la recogida de los datos generados es mucho más sencilla y estos datos son más flexibles y con mayores posibilidades que si el experimento se desarrollara de forma real, permitiendo, por ejemplo, reproducir de forma integra los recorridos monitorizados a través del entorno, lo que supone una gran ayuda a la hora de su posterior análisis.

## 6.2. Trabajo futuro

### 6.2.1. Entorno de experimentación

El entorno de experimentación desarrollado es completamente funcional pero se han detectado posibles mejoras que pueden ser implementadas. El



Figura 6.1: Ejemplo de minimapa dentro de un videojuego

entorno actual solo permite carga el modelo 3D de la Facultad de Informática, pero en su desarrollo se tuvo en cuenta que podría ser re-utilizable para cargar otros modelos, de forma que las posibilidades del entorno puedan ser fácilmente ampliadas. Por ejemplo, se podrían realizar experimentos de indicaciones en exteriores, necesitando para ello un modelo 3D que represente el entorno deseado y muy poco trabajo en el lado de la aplicación.

El entorno ha sido planteado de forma que el guía pueda en todo momento ver lo mismo que está viendo el guiado, podría plantearse otro tipo de experimento en el que esta característica no esté presente, para ello no sería necesario realizar grandes desarrollos ya que con ciertas modificaciones no muy complejas, se puede cambiar el modo de funcionamiento de la aplicación.

Por otro lado, después del estudio piloto, se detectó que podría ser de utilidad para el guía disponer de un mapa esquemático en el que se incluya la posición del guiado, de forma similar al utilizado en algunos videos juegos (ver figura 6.1), esto podría ser de gran utilidad para el guía a la hora de saber donde se encuentra la persona a la que está guiando y detectar la ruta que debe de seguir, aunque habría que tener en cuenta los efectos que esta utilidad podría provocar sobre las indicaciones generadas. Por tanto podría ser útil poder activar y desactivar ese mapa en función del experimento que se desee realizar.

### 6.2.2. Generación de logs y comunicación por red

Los logs generados por la aplicación tienen una gran utilidad y contienen mucha cantidad de información. Esto en ocasiones puede complicar el análisis de los mismos. Por tanto, puede facilitar su tratamiento la implementación de una herramienta que permita de forma automática extraer del log

la información buscada en cada uno de los momentos, por ejemplo, si solo se está analizando la conversación mantenida esa utilidad puede reconstruirla a partir de la información completa que contiene el log y presentársela al usuario. Otra característica valiosa podría ser la generación de un documento que una las conversaciones junto con las capturas realizadas durante el desarrollo de la ruta, de forma que se reproduzca el experimento a través de la información escrita y gráfica de la que se dispone.

Uno de los problemas que presenta el entorno planteado radica en la recopilación de la información generada. Es necesario extraer de cada máquina los ficheros que han sido generados en cada ejecución y esto supone una dificultad añadida. Se propone como trabajo futuro ampliar la aplicación para que sea capaz de enviar de forma automática a un servidor todos los datos generados, incluyendo logs y capturas así se garantiza que toda la información generada será recogida y se evitarán posibles pérdidas de información.

Actualmente ya existe implementada una herramienta que facilita la generación y tratamiento de logs, «Log4j». Inicialmente se desarrolló en Java pero se ha reescrito en otros lenguajes entre los que se encuentra C++, esto permitiría añadirla al código del entorno de experimentación ya desarrollado y aprovechar esta herramienta de forma que se realice el envío de logs a un Web Service creado con este objetivo.

### 6.2.3. Reproducción de sesiones anteriores

Como ya ha sido comentado en otros apartados, la información contenida en cada log permitiría reproducir toda la sesión anteriormente desarrollada. De esta forma se plantea la posibilidad de desarrollar un modo de ejecución dentro del entorno que, a través de un fichero de log, represente sobre el modelo el mismo recorrido y la misma mensajería de chat que ha sido almacenada. Esto puede facilitar el análisis de puntos complejos o recorridos complicados. Para implementar esta mejora se puede tomar como referencia el proyecto de «Martínez del Camino (2011)», donde se realiza la simulación tomando como base el log generado por otro proyecto anterior («Nuñez González (2011)») previamente estandarizado de forma que la aplicación desarrollada fuera capaz de parsearlo.

### 6.2.4. Diseño y realización de experimentos

Se ha realizado un pequeño estudio piloto para obtener una primera recogida de datos. Sería necesario realizar una serie de experimentos de mayor envergadura, de forma que la cantidad de información recogida permita afirmar la utilización de cierto tipo de expresiones a la hora de dar indicaciones. Además, el diseño que se realice para estos experimentos en función de los objetivos concretos de cada uno, puede provocar la adquisición de nuevos requisitos para implementar dentro del entorno de experimentación, mejo-

---

rando el mismo para conseguir una herramienta con grandes características dentro del análisis del lenguaje natural.



Parte I

Apéndices



## Apéndice A

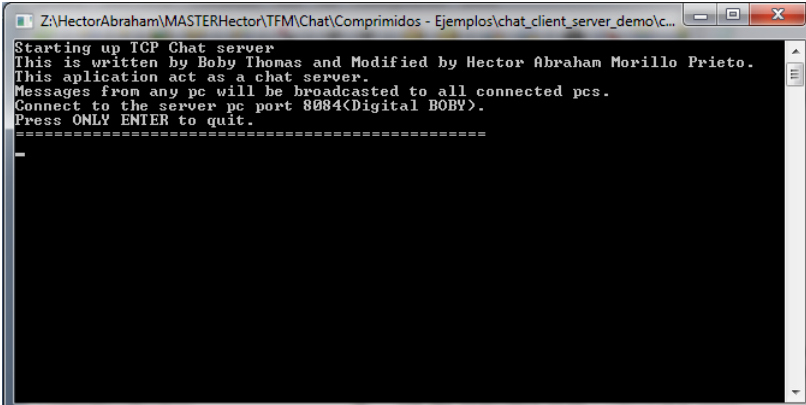
# Manual de usuario (Inicio rápido)

**RESUMEN:** En este apartado se pretende dar a los usuario de la aplicación, una guía fácil de seguir para la utilización de la misma. Hay que tener en cuenta que la aplicación hace uso de comunicación de red por lo que será necesario tener unos mínimos conocimientos de la materia.

### A.1. Inicio del servidor chat

El primer paso para arrancar todo el entorno de experimentación consiste en iniciar el servidor de chat, que permite la comunicación entre guía y guiado. Los pasos para arrancar dicho servicio son los siguientes:

- Ejecutar el fichero **chatserver.exe**



```
Z:\HectorAbraham\MASTERHector\TFM\Chat\Comprimidos - Ejemplos\chat_client_server_demo\c...
Starting up TCP Chat server
This is written by Bobby Thomas and Modified by Hector Abraham Morillo Prieto.
This application act as a chat server.
Messages from any pc will be broadcasted to all connected pcs.
Connect to the server pc port 8084(Digital BOBY).
Press ONLY ENTER to quit.
=====
-
```

Figura A.1: Servidor chat una vez que se ha iniciado correctamente

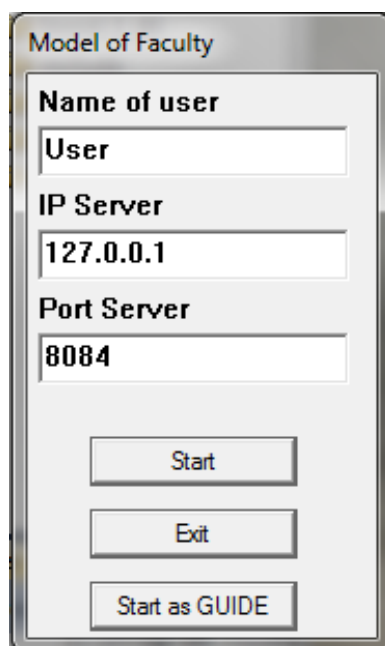
- Una vez iniciado se visualizará algo similar a la figura A.1.
- A partir de este momento ya se puede hacer uso del mismo en la aplicación. No se debe conectar más de un mismo usuario «perdido» al mismo cliente de chat, pero si pueden conectarse varios clientes «Guía». El servicio se inicia por defecto en el puerto 8084 por lo que será necesario poder realizar comunicación a través de ese puerto (Es posible que sea necesario deshabilitar fireware, antivirus o cualquier otro software que limite la capacidad de comunicación de red a través de ese puerto).
- Cuando se desee cerrar el servicio basta con presionar la tecla «Enter» dos veces y se cerrará automáticamente.

## A.2. Inicio de la aplicación

La aplicación tiene dos modos claramente diferenciados de funcionamiento, por un lado esta el modo «Guiado» que permite navegar libremente a través del modelo en tres dimensiones, además permite la comunicación a través del chat con otras instancias de la aplicación en el otro modo de funcionamiento. Por otro lado tenemos el modo de ejecución «Guía», la ventana que lanza este método de ejecución es idéntica al otro modo, en este caso el usuario no puede mover libremente la cámara sino que siempre verá lo mismo que el «Guiado» que se encuentre conectado al mismo servidor de chat. Es decir, se trata de que el «Guía» vea todo lo que se encuentra realizando el «Guiado». Además al igual que el «Guiado», el «Guía» tiene la posibilidad de comunicarse con todos los usuarios que se encuentren conectados al mismo servidor («Guiado y los diferentes guías que puedan estar conectados»).

Los pasos para arrancar la aplicación son:

- Ejecutar el fichero **Facultad3D.exe**
- Una vez se ha ejecutado dicho fichero se mostrará una ventana como la de la figura A.2.
- En esa pantalla se puede introducir un nombre de usuario personalizado, que será el identificador para los mensajes enviados a través del chat. La dirección IP en la que se encuentra el servidor del chat y el puerto.
- Cuando se han completado los campos anteriores la aplicación puede ser iniciada en dos versiones, por un lado como la persona que se encuentra perdida «START», en este modo el usuario podrá mover libremente la cámara por el modelo. También se puede iniciar como guía «START AS GUIDE», en este modo el usuario verá en su pantalla lo mismo que se encuentra viendo la persona perdida conectada



The image shows a configuration dialog box titled "Model of Faculty". It contains three input fields for user information:

- Name of user:** The input field contains the text "User".
- IP Server:** The input field contains the IP address "127.0.0.1".
- Port Server:** The input field contains the port number "8084".

Below the input fields are three buttons:

- Start**
- Exit**
- Start as GUIDE**

Figura A.2: Configuración inicial aplicación



Figura A.3: Vista general de la aplicación navegando por el modelo 3D

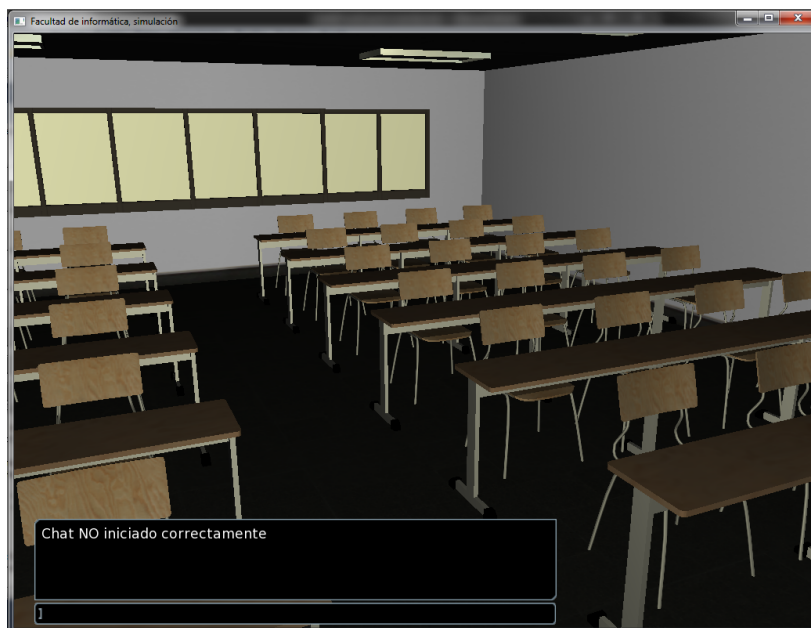


Figura A.4: Vista general de la aplicación con el chat

al mismo servidor, de esta forma podrá, a través del chat, guiar o dar indicaciones a la otra persona (Figura A.3).

- Dentro de la aplicación, para poder mostrar u ocultar la ventana del chat, basta con pulsar la tecla «TAB» (tabulador), en la figura A.4 se puede ver el chat de la aplicación.
- Es importante si se está realizando en experimento o bien se desea tener la funcionalidad del chat, comprobar que cuando se ha iniciado la aplicación el primer mensaje que se observa en la ventana confirma la correcta inicialización y conexión con el chat. Si por el contrario el mensaje que aparece es el mismo que en la imagen anterior, no se ha podido iniciar el chat y por tanto no se dispondrá de esa funcionalidad. (Si solo se desea realizar una navegación por el modelo, no es necesario iniciar el chat).

### A.3. Navegación por la aplicación

Para navegar a través del modelo se deben utilizar las flechas del teclado junto con el ratón para dirigir los movimientos y la dirección de la cámara.

También existe la posibilidad de colocar la cámara en un punto determinado del modelo, esta funcionalidad puede ser usada para realizar algunos experimentos. Para utilizar esta funcionalidad basta con introducir en el chat

---

la siguiente instrucción: «PUTMEIN,XXX,YYY,ZZZ», donde XXX, YYY y ZZZ son las coordenadas del punto en el que se quiere situar la cámara. Como es lógico esta funcionalidad solo puede ser utilizada en una instancia de la aplicación arrancada en modo «Guiado».



## Apéndice B

# Posible rutas para experimento

**RESUMEN:** Este apartado contiene una serie de rutas diseñadas para los experimentos iniciales, se trata de rutas sencillas y relativamente fáciles de realizar cuyo objetivo es el de verificar el correcto funcionamiento del entorno de experimentación así como unos adquirir unos resultados iniciales que permitan sacar algunas conclusiones para mejorar o continuar con la investigación. A través de estas rutas se trata también de obtener unas primeras ideas sobre la forma de expresión que utilizamos habitualmente así como detectar posibles errores que hayan podido encontrarse ocultos hasta este momento.

### B.1. Ruta 1

**Origen:** Puerta principal de la facultad

**Destino:** Despacho 411 (O cualquier otro despacho de la cuarta planta)

**Posible interés de la ruta:** Al iniciar la ruta desde la puerta principal es necesario pasar por el hall de entrada que tiene diferentes posibilidades, posiblemente sea uno de los puntos más complicados para realizar las indicaciones dentro de la facultad. El guía deberá indicar una forma para subir las plantas necesarias hasta llegar a la 4<sup>o</sup> planta donde se encuentra el despacho, un punto con cierta complejidad también será indicar que despacho es exactamente el que tiene que tiene como destino, por tanto será necesario identificar un elemento entre varios similares.

**Coordenadas del punto inicial:**

- X: 1987
- Y: 797
- Z: 2099

## B.2. Ruta 2

**Origen:** Sala de juntas (o viceversa)

**Destino:** Salón de actos (o viceversa)

**Posible interés de la ruta:** La sala de juntas se encuentra un poco apartada, por tanto, la ruta tiene sentido en ambas direcciones, para llegar a ellas es necesario atravesar el hall principal con las dificultades que ese recinto presenta dadas sus características. Ambas salas se encuentran en el mismo lado del edificio aunque en plantas diferentes, los problemas que pueden tener esta ruta vienen dados por la situación de las salas, ya que no son visibles rápidamente y son necesarias ciertas indicaciones para llegar a ellas.

**Coordenadas sala de juntas:**

- X: 2079
- Y: 1181
- Z: 1323

**Coordenadas salón de actos:**

- X: 3001
- Y: 771
- Z: 1826

## B.3. Ruta 3

**Origen:** Despacho 420-bis

**Destino:** Barra cafetería

**Posible interés de la ruta:** El desarrollo de esta ruta tiene varias posibilidades, en función de la ruta que decida realizar el guía podrá encontrar una serie de dificultades u otras. Puede ser interesante ver que ruta decide indicar en cada caso el guía y que técnicas utiliza. La cafetería es una zona con cierta complejidad ya que tiene dos posibles entradas, cada una de ellas a un lado de la facultad y con características diferentes. Con esta ruta se podría tratar de identificar si todos los guías deciden llevar al guiado por el mismo recorrido y utilizar las mismas referencias o por el contrario cada uno realiza unas diferentes.

**Coordenadas despacho 420-bis:**

- X: -203
- Y: 2013
- Z: -151

## B.4. Ruta 4

**Origen:** Barra cafetería

**Destino:** Laboratorio 7

**Posible interés de la ruta:** La cafetería es un punto complicado, tiene varias salidas y cada una de ellas da a una parte totalmente diferente de la facultad. Por otro lado el pasillo en el que se encuentran todos los laboratorios es muy similar, el guía tendrá que buscar la forma de indicar el laboratorio exacto al que se dirige, para ello tendrá que apoyarse en el mobiliario o otras características propias del pasillo.

**Coordenadas barra cafetería:**

- X: -1535
- Y: 879
- Z: 1754

## B.5. Ruta 5

**Origen:** Puerta principal de la facultad

**Destino:** Despacho del Decano (Decanato)

**Posible interés de la ruta:** La ruta tiene bastantes cambios de dirección y giros, puede ser interesante para después poder analizar esas indicaciones. Además en esta ruta se pasa a través del hall con la dificultad añadida que este plantea, ya que existen diferentes posibilidades para salir del mismo y es necesario realizar una correcta indicación. Dentro del decanato deberá ser capaz de indicar cuál de los despachos es el del decano.

**Coordenadas de la puerta principal:**

- X: 1987
- Y: 797
- Z: 2099



# Apéndice C

## Ejemplos de Logs

### RESUMEN:

En este apéndice se muestran muestras unos fragmentos de los logs obtenidos en las simulaciones del experimento piloto comentado en el apartado «4.3». Solo se muestra un fragmento debido a que los logs generados tienen miles de líneas. La intención es hacer ver como se realiza el almacenamiento de las posiciones por las que va pasando el usuario así como toda la mensajería que se intercambia entre los usuarios (Guía y Guiado). Como se va a poder ver en las siguientes secciones ambos logs son muy similares, aunque cada uno de ellos sea generado por un modo de funcionamiento de la aplicación, con esto lo que se pretende es que una sesión de pruebas pueda ser recreada aunque solo se disponga del log generado por una parte, se ha planteado así como una medida de seguridad frente a la pérdida de datos en caso de que se produzca algún error en la ejecución de una parte.

Como ya se ha comentado en otros apartados lo realmente interesante del este entorno de monitorización es la capacidad de almacenamiento de todas las conversaciones mantenidas a través del chat, además como se puede observar en los log de las secciones siguientes cada vez que un usuario hace uso del chat, es decir, envía un mensaje, se realiza una captura de lo que se encuentra viendo en ese momento, esto puede permitir conocer los lugares donde ha tenido dudas, o las indicaciones del guía no han sido suficientes. Con esta información podemos detectar que tipo de expresiones son menos eficaces a la hora de indicar como alcanzar un objetivo, así como las zonas que son más complicadas para ser descritas o entendidas por los usuarios.

### C.1. Log generado por un usuario en modo Guiado

– USER –

```
setAttributes ->User: Mandado 47; IP: 169.254.22.37; Port: 876097592  
Finalización funcion init  
Conexión del chat Correcta - Conectado!
```

Creado hilo de lecturaMandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.875151,-0.0980578,0.473823)

Mandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.850465,-0.113677,0.513616)

Mandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.835966,-0.121477,0.535182)

Mandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.8162,-0.131865,0.562536)

Mandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.788648,-0.147419,0.596926)

Mandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.752484,-0.178414,0.634)

Mandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.710951,-0.216907,0.668971)

Mandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.679697,-0.222015,0.699103)

...

Mandado 47:CameraPosition:(1990.39,795.276,2117.18)CameraDirection(0.1826,-0.092845,0.978818)

Mandado 47:CameraPosition:(1990.39,795.274,2117.2)CameraDirection(0.1826,-0.092845,0.978818)

Mandado 47:CameraPosition:(1990.4,795.273,2117.21)CameraDirection(0.1826,-0.092845,0.978818)

Mandado 47:CameraPosition:(1990.4,795.273,2117.21)CameraDirection(0.1826,-0.092845,0.978818)

Mandado 47:CameraPosition:(1990.4,795.273,2117.22)CameraDirection(0.1826,-0.092845,0.978818)

Guia 47:-203.9, 2013.6, -151.4

Guia 47:son las coordenadas del punto de partida

File: captures/28-5-13-14-1-28.jpg- CameraPosition:( 1990.4, 795.273, 2117.22)

PUTMEIN,-203,2013,-151

Mandado 47:CameraPosition:(-203,2013,-151)CameraDirection(0.1826,-0.092845,0.978818)

Mandado 47:CameraPosition:(-203,2013,-151)CameraDirection(0.0323642,-0.035374,0.998874)

Mandado 47:CameraPosition:(-203,2013,-151)CameraDirection(-0.0645587,0.00127165,0.997937)

Mandado 47:CameraPosition:(-203,2013,-151)CameraDirection(-0.082806,-0.00658252,0.996568)

Mandado 47:CameraPosition:(-203,2013,-151)CameraDirection(-0.0932146,-0.0118184,0.995601)

Mandado 47:CameraPosition:(-203,2013,-151)CameraDirection(-0.0145084,-0.0850221,0.996298)

...

Mandado 47:CameraPosition:(-207.021,2007.32,-117.371)CameraDirection(-0.996677,-0.0641356,0.0507138)

Mandado 47:CameraPosition:(-207.021,2007.32,-117.371)CameraDirection(-0.998773,-0.0432214,-0.0251298)

Mandado 47:CameraPosition:(-207.021,2007.32,-117.371)CameraDirection(-0.997701,-0.0379895,-0.0565197)

Guia 47:sal del despacho, hasta el pasillo

Mandado 47:CameraPosition:(-208.043,2007.28,-117.429)CameraDirection(-0.997701,-0.0379895,-0.0565197)

Mandado 47:CameraPosition:(-211.58,2007.14,-117.629)CameraDirection(-0.997701,-0.0379895,-0.0565197)

Mandado 47:CameraPosition:(-219.009,2006.86,-118.05)CameraDirection(-0.997701,-0.0379895,-0.0565197)

...

Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.993824,-0.0850126,-0.0717994)

Mandado 47: CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.994357,-0.0850126,-0.0639917)

Guia 47:sal al pasillo principal

Mandado 47: CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.989355,-0.124067,0.0765736)

Mandado 47: CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.892205,-0.155177,0.424259)

...

Mandado 47: CameraPosition:(-512.392,1980.97,311.591)CameraDirection(-0.996987,0.0510123,-0.0593718)

Mandado 47: CameraPosition:(-512.392,1980.97,311.591)CameraDirection(-0.997139,0.0510123,-0.0567615)

Guia 47:camina de frente hasta el fondo sin cruzar esas puertas

Mandado 47: CameraPosition:(-513.024,1981.01,311.555)CameraDirection(-0.997139,0.0510123,-0.0567615)

Mandado 47: CameraPosition:(-516.506,1981.18,311.357)CameraDirection(-0.997139,0.0510123,-0.0567615)

Uno de los puntos ha destacar de este log es la utilización de la instrucción «PUTMEIN,-203,2013,-151» que permite colocar al usuario perdido en las coordenadas descritas, estas corresponde con el origen de las una de las rutas planteadas en el experimento piloto para comprobar el correcto funcionamiento del entorno de experimentación y obtener unos primeros resultados.

Otro punto importante es la similitud entre el log de está sección y el que se verá en la sección siguiente, como se puede ver se podría reconstruir la conversación unicamente con uno de ellos así como el recorrido realizado, aunque no ocurriría lo mismo con las capturas realizadas en la otra instancia de la aplicación, ya que los ficheros se almacenan en el disco local de la maquina que está ejecutando el proyecto.

## C.2. Log generado por un usuario en modo Guía

El log generado es muy similar al del modo anterior.

– GUIDE –

setAttributes -> User: Guia 47; IP: 127.0.0.1; Port: 876097592

Finalización funcion init

Conexión del chat Correcta - Conectado!

Creado hilo de lectura

Mandado 47: CameraPosition:(1987,797,2099)CameraDirection(0.875151,-0.0980578,0.473823)

Mandado 47: CameraPosition:(1987,797,2099)CameraDirection(0.850465,-0.113677,0.513616)

Mandado 47: CameraPosition:(1987,797,2099)CameraDirection(0.835966,-0.121477,0.535182)

Mandado 47: CameraPosition:(1987,797,2099)CameraDirection(0.8162,-0.131865,0.562536)

Mandado 47: CameraPosition:(1987,797,2099)CameraDirection(0.788648,-0.147419,0.596926)

Mandado 47: CameraPosition:(1987,797,2099)CameraDirection(0.752484,-0.178414,0.634)

Mandado 47: CameraPosition:(1987,797,2099)CameraDirection(0.710951,-0.216907,0.668971)

Mandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.679697,-0.222015,0.699103)  
 Mandado 47:CameraPosition:(1987,797,2099)CameraDirection(0.651726,-0.222015,0.72525)

...

Mandado 47:CameraPosition:(1990.4,795.273,2117.22)CameraDirection(0.1826,-  
 0.092845,0.978818)  
 File: captures/28-5-13-14-1-7.jpg- CameraPosition:( 1990.4, 795.273, 2117.22)Guia  
 47:-203.9, 2013.6, -151.4)  
 File: captures/28-5-13-14-1-17.jpg- CameraPosition:( 1990.4, 795.273, 2117.22)Guia  
 47:son las coordenadas del punto de partida  
 Mandado 47:CameraPosition:(-203,2013,-151)CameraDirection(0.1826,-0.092845,0.978818)  
 Mandado 47:CameraPosition:(-203,2013,-151)CameraDirection(0.0323642,-0.035374,0.998874)

...

File: captures/28-5-13-14-1-43.jpg- CameraPosition:( -207.021, 2007.32, -117.371)Guia  
 47:sal del despacho, hasta el pasillo  
 Mandado 47:CameraPosition:(-208.043,2007.28,-117.429)CameraDirection(-0.997701,-  
 0.0379895,-0.0565197)

...

Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.994516,-  
 0.0902288,-0.0535331)  
 Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.994296,-  
 0.087621,-0.0613656)  
 Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.994008,-  
 0.0850126,-0.0691973)  
 Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.993824,-  
 0.0850126,-0.0717994)  
 Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.994357,-  
 0.0850126,-0.0639917)  
 File: captures/28-5-13-14-1-51.jpg- CameraPosition:( -540.054, 1990.13, -109.989)Guia  
 47:sal al pasillo principal  
 Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.989355,-  
 0.124067,0.0765736)  
 Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.892205,-  
 0.155177,0.424259)  
 Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.730244,-  
 0.165514,0.662927)  
 Mandado 47:CameraPosition:(-540.054,1990.13,-109.989)CameraDirection(-0.446281,-  
 0.168096,0.879043)

...

Mandado 47:CameraPosition:(-512.392,1980.97,311.591)CameraDirection(-0.996987,0.0510123,-  
 0.0593718)  
 Mandado 47:CameraPosition:(-512.392,1980.97,311.591)CameraDirection(-0.997139,0.0510123,-



Figura C.1: Captura de imagen generada en una de las instrucciones - 28-5-13-14-1-51.jpg

0.0567615)

File: captures/28-5-13-14-2-25.jpg- CameraPosition:( -512.392, 1980.97, 311.591)Guia 47:camina de frente hasta el fondo sin cruzar esas puertas

Mandado 47: CameraPosition:(-513.024,1981.01,311.555)CameraDirection(-0.997139,0.0510123,-0.0567615)

Mandado 47: CameraPosition:(-516.506,1981.18,311.357)CameraDirection(-0.997139,0.0510123,-0.0567615)

Como se puede ver en el log la captura con nombre «28-5-13-14-1-51.jpg - C.1» se genera para la instrucción «Guia 47:sal del despacho, hasta el pasillo». Está imagen ayuda por tanto a la comprensión del lugar en el que se encuentra la persona guiada y que es lo que está viendo en ese momento concreto.



# Bibliografía

- MARTÍNEZ DEL CAMINO, G. *Simulación de emergencia en entornos 3D*. Facultad de Infomática - Universidad Complutense de Madrid, 2011.
- MARTÍNEZ DEL CAMINO, G. *Simulaciones de situaciones de emergencia en entornos 3D*. Facultad de Infomática - Universidad Complutense de Madrid, 2012.
- CLIPPINGER, J. A. *A discourse speaking program as a preliminary theory of discourse behaviour and a limited theory of psychoanalytic discourse*. Tesis Doctoral, University of Pennsylvania, EEUU, 1974.
- COHÉN, P. y PERRAULT, C. Elements of a plan-based theory of speechacts. *CognitiveScience*, vol. 3(1), 1979. Disponible en [http://www.cs.rochester.edu/u/kautz/Courses/577autumn2007/elements\\_of\\_plan\\_based\\_theory\\_speech\\_acts\\_cohen.pdf](http://www.cs.rochester.edu/u/kautz/Courses/577autumn2007/elements_of_plan_based_theory_speech_acts_cohen.pdf) (último acceso, Marzo, 2013).
- CRUZ RUIZ, F., DURBEY CARRASCO, A. y SANZ BRIONES, J. *Entorno virtual 3D multiusuario para simulación de escenarios de evacuación*. Facultad de Infomática - Universidad Complutense de Madrid, 2011.
- DIONNE, D., PUENTE, S. D. L., LEÓN, C., HERVÁS, R. y GERVÁS, P. A model for human readable instruction generation using level-based discourse planning and dynamic inference of attributes. *In: Proceedings of the 12th European Workshop on Natural Language Generation (ENLG)*, páginas 66–73, 2009.
- GOLDBERG, E., DRIEDGER, N. y KITTREDGE, R. Using natural-language processing to produce weather forecast. *IEEE Expert*, vol. 9(2), 1994. Disponible en <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=294135> (último acceso, Marzo, 2013).
- NUÑEZ GONZÁLEZ, E. *Modelo de evacuación de multitudes mediante agentes y transcripción de comportamiento*. Facultad de Infomática - Universidad Complutense de Madrid, 2011.
- LEÓN, C., PUENTE, S., DIONNE, D., HERVÁS, R. y GERVÁS, P. Constructing world abstractions for natural language in virtual 3d environments. vol. 226, páginas 77–86, 2009.
- MANN, W. C. y THOMPSON, S. A. Rhetorical structure theory: towards a functional theory of text organization. *Text*, vol. 8(3), 1988. Disponible en <http://www.cis.upenn.edu/~nenkova/Courses/cis700-2/rst.pdf> (último acceso, Marzo, 2013).

- McKEOWN, K. *Text generation: Using strategies and focus constraints to generate natural language text*. Cambridge University Press, 1985.
- PÉREZ Y PÉREZ, R. *Mexica : a computer model of creativity in writing*. Tesis Doctoral, University of Sussex, UK, 1999.
- PÉREZ Y PÉREZ, R. y SHARPLES, M. Mexica: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 13(2), páginas 119–139, 2001. Disponible en [http://www.eee.bham.ac.uk/sharplem/papers/mexica\\_jetai.pdf](http://www.eee.bham.ac.uk/sharplem/papers/mexica_jetai.pdf) (último acceso, Marzo, 2013).
- REITER, E. y DALE, R. *Building Natural Language Generation Systems*. Cambridge University Press, 2000. ISBN 0 521 62036 8.
- REITER, E., ROBERTSON, R. y OSMAN, L. Knowledge acquisition for natural language generation. *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, páginas 389–399, 1999. Disponible en [ac1.ldc.upenn.edu/W/W00/W00-1429.pdf](http://ac1.ldc.upenn.edu/W/W00/W00-1429.pdf) (último acceso, Marzo, 2013).
- SHARPLES, M. *How we write: Writing as Creative Design*. Routledge, London, 1999.